How long, O Bayesian network, will I sample thee?

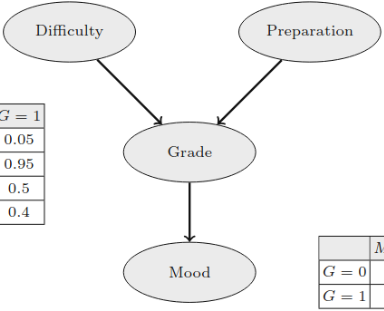
A program analysis perspective on expected sampling times

A paper for **ICALP 2018** by
Kevin Batz
Benjamin Lucien Kaminski
Joost-Pieter Katoen
Christoph Matheja

promoted by **Márk Széles** (me)

Bayesian networks

$D=0$	$D=1$
0.6	0.4



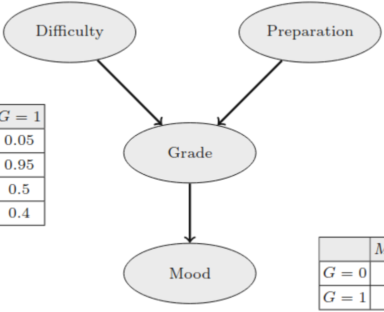
$P=0$	$P=1$
0.7	0.3

	$G=0$	$G=1$
$D=0,\ P=0$	0.95	0.05
$D=1,\ P=1$	0.05	0.95
$D=0,\ P=1$	0.5	0.5
$D=1,\ P=0$	0.6	0.4

	$M=0$	$M=1$
$G=0$	0.9	0.1
$G=1$	0.3	0.7

Bayesian networks

$D=0$	$D=1$
0.6	0.4



$P=0$	$P=1$
0.7	0.3

	$G=0$	$G=1$
$D=0,\ P=0$	0.95	0.05
$D=1,\ P=1$	0.05	0.95
$D=0,\ P=1$	0.5	0.5
$D=1,\ P=0$	0.6	0.4

	$M=0$	$M=1$
$G=0$	0.9	0.1
$G=1$	0.3	0.7

An example inference problem

What is the probability that a student prepares, but ends up in a bad mood?

$$P(M=0|P=1) \quad = \quad ???$$

How to solve inference problems?

Exact inference

- Gives an exact result
- **N**ot **P**leasant to compute (NP-hard)

How to solve inference problems?

Exact inference

- Gives an exact result
- **N**ot **P**leasant to compute (NP-hard)

Sampling-based methods

- Draw random samples from distribution.
- Use the samples to compute an approximate result.
- This is **often** faster.

Rejection sampling

An example inference problem

What is the probability that a student prepares, but ends up in a bad mood?

$$P(M = 0|P = 1) \quad = \quad ???$$

Collect samples using a loop:

```
repeat {
   (D,P,G,M):≈ μ
} until (P = 1)
```

Result:

D	0	1	0	1
P	1	1	1	1
G	1	1	1	0
M	1	1	1	0

$$P(M = 0|P = 1) = 1/4$$

“How long, O Bayesian network, will I sample thee?”

The problem

Compute how long it takes to get a sample via rejection sampling.

“How long, O Bayesian network, will I sample thee?”

The problem

Compute how long it takes to get a sample via rejection sampling.

The solution

1. Convert the Bayesian network the an imperative, probabilistic program.

"How long, O Bayesian network, will I sample thee?"

The problem

Compute how long it takes to get a sample via rejection sampling.

The solution

1. Convert the Bayesian network the an imperative, probabilistic program.
2. The rejection sampling query can now be formulated as a loop.

“How long, O Bayesian network, will I sample thee?”

The problem

Compute how long it takes to get a sample via rejection sampling.

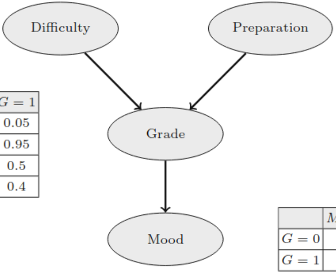
The solution

1. Convert the Bayesian network the an imperative, probabilistic program.
2. The rejection sampling query can now be formulated as a loop.
3. Reason about the expected runtime of the loop using a weakest precondition-style calculus.

Rejection sampling queries as programs

$D=0$	$D=1$
0.6	0.4

$P=0$	$P=1$
0.7	0.3



	$G=0$	$G=1$
$D=0,\ P=0$	0.95	0.05
$D=1,\ P=1$	0.05	0.95
$D=0,\ P=1$	0.5	0.5
$D=1,\ P=0$	0.6	0.4

	$M=0$	$M=1$
$G=0$	0.9	0.1
$G=1$	0.3	0.7

```
repeat {
    x_D :≈ 0.6·⟨0⟩ + 0.4·⟨1⟩;
    x_P :≈ 0.7·⟨0⟩ + 0.3·⟨1⟩;
    if (x_D = 0 ∧ x_P = 0) {
        x_G :≈ 0.95·⟨0⟩ + 0.05·⟨1⟩
    } else if (x_D = 1 ∧ x_P = 1) {
        x_G :≈ 0.05·⟨0⟩ + 0.95·⟨1⟩
    } else if (x_D = 0 ∧ x_P = 1) {
        x_G :≈ 0.5·⟨0⟩ + 0.5·⟨1⟩
    } else {
        x_G :≈ 0.6·⟨0⟩ + 0.4·⟨1⟩
    };
    if (x_G = 0) {
        x_M :≈ 0.9·⟨0⟩ + 0.1·⟨1⟩
    } else {
        x_M :≈ 0.3·⟨0⟩ + 0.7·⟨1⟩
    }
} until (x_P = 1)
```

A taste of the ert-calculus

- Let Σ be the set of possible program states. The expected runtime transformer has type

$$\mathrm{ert}[-] : \mathrm{Progs} \to (\Sigma \to \mathbb{R}^{\infty}_{\geq 0}) \to (\Sigma \to \mathbb{R}^{\infty}_{\geq 0}).$$

A taste of the ert-calculus

- Let Σ be the set of possible program states. The expected runtime transformer has type

$$\mathrm{ert}[-] : \mathrm{Progs} \to (\Sigma \to \mathbb{R}_{\geq 0}^{\infty}) \to (\Sigma \to \mathbb{R}_{\geq 0}^{\infty}).$$

- Compare with weakest precondition:

$$\mathrm{wp}[-] : \mathrm{Progs} \to (\Sigma \to 2) \to (\Sigma \to 2).$$

A taste of the ert-calculus

- Let Σ be the set of possible program states. The expected runtime transformer has type

$$\mathrm{ert}[-] : \mathrm{Progs} \to (\Sigma \to \mathbb{R}^{\infty}_{\geq 0}) \to (\Sigma \to \mathbb{R}^{\infty}_{\geq 0}).$$

- Compare with weakest precondition:

$$\mathrm{wp}[-] : \mathrm{Progs} \to (\Sigma \to 2) \to (\Sigma \to 2).$$

- There are rules similar to weakest precondition.

Why choose this paper?

- It is accessible.
- You can learn about probabilistic programming.
- You can learn about verification of probabilistic programs.
- The paper is a nice mix of theory and practice.
- There is a wide variety of possible second papers.