An Algebraic Theory of Conditioning

Dario Stein, Sam Staton

University of Oxford

LAFI'21
How to express dependence on data?

Scoring

\[
\text{position} = \text{normal}(0, 100) \\
\text{observe}(\text{normal(position, 5)}, 42)
\]

\[
\text{score} (pdf\_normal(position, 5)(42))
\]

vs

Exact Conditioning

\[
\text{# generative model} \\
\text{position} = \text{normal}(0, 100) \\
\text{measurement} = \text{normal(position, 5)}
\]

\[
\text{# conditioning} \\
\text{measurement} =:= 42
\]

[Stan, WebPPL] [Hakaru, Infer.NET]
Advantages of Exact Conditioning

- Modularity

```python
ys = gp_sample(n=100, kernel=rbf)
for (i, obs) in observations:
    ys[i] =:= obs
```

Scoring statements would need to be interleaved with `gp_sample`
Advantages of Exact Conditioning

- Intuitiveness & Correctness

\[ x = \text{normal}(0,1) \]
\[ y = \text{normal}(0,1) \]
\[ x =:= y \]

\[ x = \text{normal}(0,1) \]
\[ y = \text{normal}(0,1) \]
\[ \text{observe} \left( \text{normal}(0,0.01), x-y \right) \]

# x = y does not hold!
# why 0.01?

\[ x = \text{normal}(0,\sqrt{1/2}) \]
\[ y = x \]

# x = y holds exactly

Closest approximation using scoring …
A language for exact conditioning

- Toy language: Only Gaussians & affine maps + conditioning
  - Kalman filters, Ridge regression, Gaussian processes
- Reference implementation
  - Calling `normal(\mu, \sigma)` allocates a latent RV
  - Maintain a joint prior over all RVs
  - When conditioning, update the prior
    - Symbolic inference: Gaussians are self-conjugate
Verifying properties

Commutativity

\[ A_1 =: A_2 \]
\[ B_1 =: B_2 \]
\[ \approx \]

Substitutivity

\[ x =: y ; c[x] \]
\[ \approx \]

Equivalent conditions

\[ 2x =: -4y + 2 \]
\[ \approx \]

MIND BOREL’S PARADOX!

\[ x/y =: 1 \]

[Shan]
Hard questions

• How to generalize to a non-toy language?
• Which nice behavior transfers?
• What should the general properties of (=:=) be?

WANTED:

General\(^1\) compositional\(^2\) semantics for exact conditioning
I. General

• Exact conditioning on continuous variables is hard
  • Borel’s paradox & [Jules Jacobs, POPL’21]

• Conditioning is about ...
  • Densities ✗
  • Limits ✗
  • Measure Theory ✗
  • **Universal property → Markov categories** [Fritz, Cho & Jacobs] ✓
II. Compositional

• Markov category conditionals are still a transformation of whole (closed) programs

• Cond-construction: Explain equivalence of open programs

\[ x |\leftarrow \text{let } y = \text{normal}(0,1) \text{ in } x =:= y; \text{ return } (x, y) \]

Markov categories + Cond-construction = Compositional Exact Conditioning
Summary

• Language for Gaussian conditioning with good properties
• Those good properties generalize!
  • Conditioning via universal properties
• Markov categories + Cond = Compositional Exact Conditioning
  • Denotational semantics for symbolic disintegration [Shan]
• Study well-behaved Markov categories!
• We can fully axiomatize the Gaussian language!

• The only things you need to know is
  • The language is commutative & discardable
  • IID Gaussians are invariant under rotations
  • Nice laws for conditioning

\[ a, b \mid \varphi : 1 \vdash (a = b); \varphi[a] \equiv (a = b); \varphi[b] \quad (10) \]
\[ a, b \mid \varphi : 1 \vdash (a = b); \forall x.\varphi[x] \equiv \forall x.(a = b); \varphi[x] \quad (11) \]
\[ - \mid \varphi : 1 \vdash \forall x.(x = c); \varphi[x] \equiv \varphi[c] \quad (12) \]

**BBonus:** There is no Borel’s paradox in Gaussian probability