Toward Interactive Statistical Modeling

Sooraj Bhat[1] Ashish Agarwal[2] Alexander Gray[1] Richard Vuduc[1]

[1] College of Computing, Georgia Institute of Technology
[2] Department of Computer Science, Yale University

Workshop on Automated Program Generation for Computational Science
at ICCS 2010

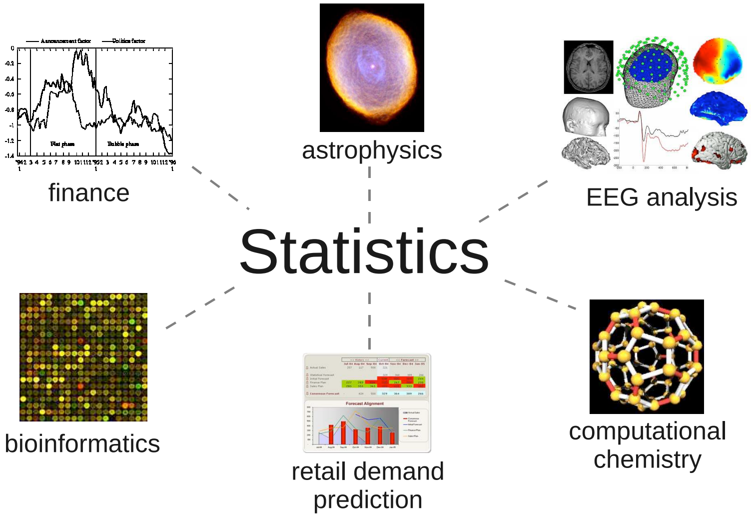
The team

Ashish Agarwal
Yale University
programming languages, optimization

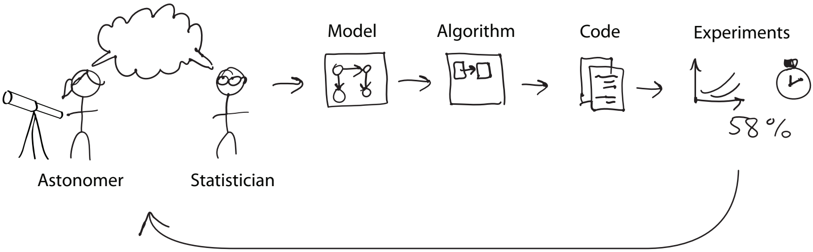
Alexander Gray
Georgia Institute of Technology
machine learning, optimization

Richard Vuduc
Georgia Institute of Technology
high-performance computing, linear algebra

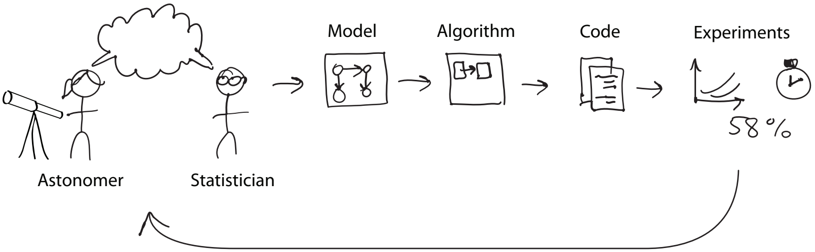
Statistics is everywhere



But the workflow is still mostly manual

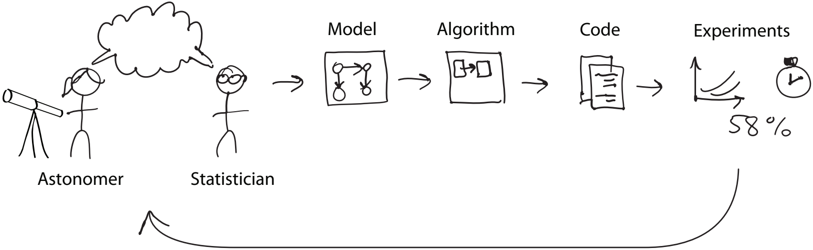


But the workflow is still mostly manual



Vision: reduce development time while retaining correctness & efficiency.

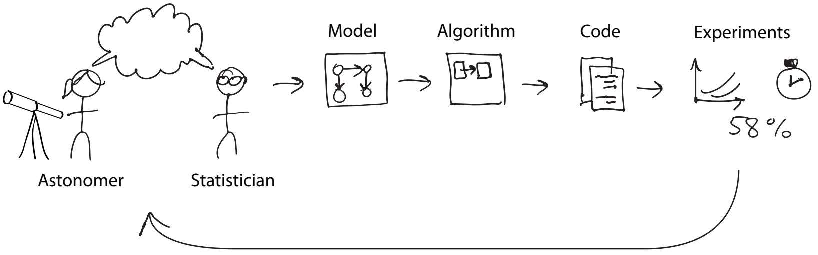
But the workflow is still mostly manual



Vision: reduce development time while retaining correctness & efficiency.

- **key ideas:** mechanization, type theory

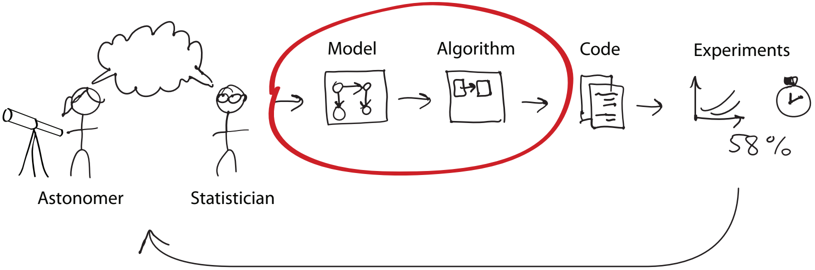
But the workflow is still mostly manual



Vision: reduce development time while retaining correctness & efficiency.

- **key ideas:** mechanization, type theory
- **key contribution:** first rigorous symbolic formalization of statistics

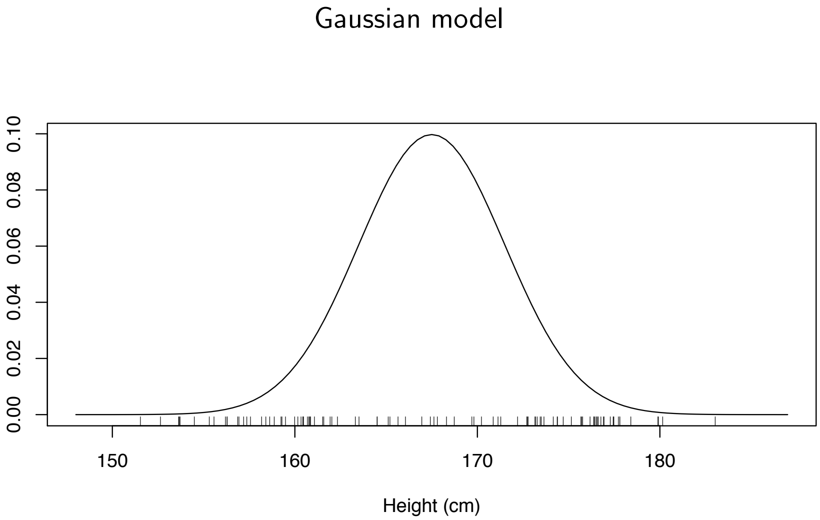
But the workflow is still mostly manual



Vision: reduce development time while retaining correctness & efficiency.

- **key ideas:** mechanization, type theory
- **key contribution:** first rigorous symbolic formalization of statistics

Example: Modeling height data



Trying alternative statistical models

Formulation:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Trying alternative statistical models

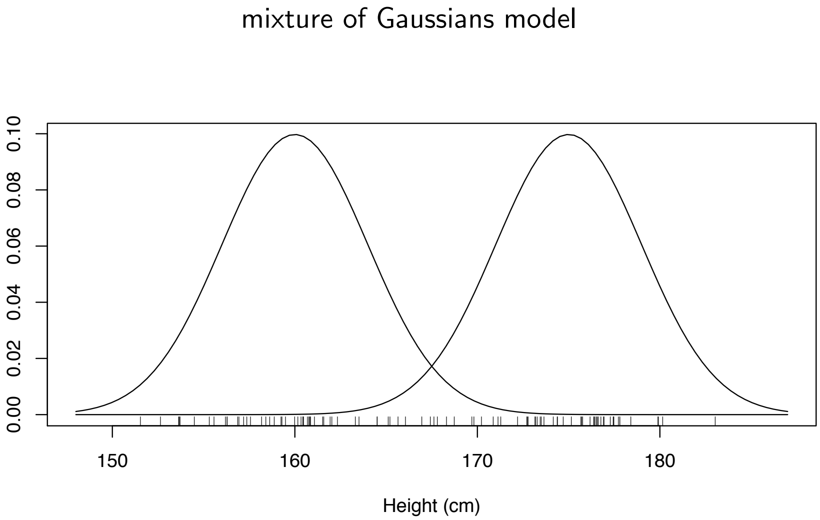
Formulation:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *closed form*

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Example: Modeling height data



Trying alternative statistical models

Formulation:

$$X_i \sim \mathrm{Normal}(\theta, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *closed form*

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Formulation:

$$Z_i \sim \mathrm{Bernoulli}(0.5)$$
$$X_i \sim \mathrm{Normal}((1 - Z_i)\theta_0 + Z_i\theta_1, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Trying alternative statistical models

Formulation:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *closed form*

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Formulation:

$$Z_i \sim \text{Bernoulli}(0.5)$$
$$X_i \sim \text{Normal}((1 - Z_i)\theta_0 + Z_i\theta_1, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *an EM algorithm (type of optimizer)*

```
(θ̂0, θ̂1) := rand();
while (...)
   for i = 1 to n do
      γi  :=  φ(xi; θ̂1,1)/(φ(xi; θ̂0,1)+φ(xi; θ̂1,1));
   θ̂0  :=  Σ_{i=1}^n (1-γi)*xi  /  Σ_{i=1}^n (1-γi);
   θ̂1  :=  Σ_{i=1}^n  γi*xi  /  Σ_{i=1}^n  γi;
return (θ̂0, θ̂1);
```

Trying alternative statistical models

Formulation:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *closed form*

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Formulation:

$$Z_i \sim \text{Bernoulli}(0.5)$$
$$X_i \sim \text{Normal}((1 - Z_i)\theta_0 + Z_i\theta_1, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *an EM algorithm (type of optimizer)*

```
(θ̂0, θ̂1) := rand();
while (...)
    for i = 1 to n do
        γi := φ(xi; θ̂1,1)/(φ(xi; θ̂0,1)+φ(xi; θ̂1,1));
    θ̂0 := Σ_{i=1}^n (1-γi)*xi / Σ_{i=1}^n (1-γi);
    θ̂1 := Σ_{i=1}^n γi*xi / Σ_{i=1}^n γi;
return (θ̂0, θ̂1);
```

- Minor conceptual changes lead to vastly different algorithms.

Trying alternative statistical models

Formulation:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *closed form*

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Formulation:

$$Z_i \sim \text{Bernoulli}(0.5)$$
$$X_i \sim \text{Normal}((1 - Z_i)\theta_0 + Z_i\theta_1, 1)$$
$$\hat{\theta} = \arg\max_{\theta} f(x \mid \theta)$$

Solution: *an EM algorithm (type of optimizer)*

```
(θ̂₀, θ̂₁) := rand();
while (...)
   for i = 1 to n do
      γᵢ := φ(xᵢ; θ̂₁,1)/(φ(xᵢ; θ̂₀,1)+φ(xᵢ; θ̂₁,1));
   θ̂₀ := Σᵢ₌₁ⁿ (1-γᵢ)*xᵢ / Σᵢ₌₁ⁿ (1-γᵢ);
   θ̂₁ := Σᵢ₌₁ⁿ γᵢ*xᵢ / Σᵢ₌₁ⁿ γᵢ;
return (θ̂₀, θ̂₁);
```

- Minor conceptual changes lead to vastly different algorithms.
- AutoBayes (2003) handles varying models for MLE/MAP – but there are many more possible axes of variation.

Idea: Let's mechanize these derivations

Goal: declarative specification $\longrightarrow$ executable code

Idea: Let's mechanize these derivations

Goal: declarative specification $\longrightarrow$ executable code

$\Rightarrow$ **need a symbolic representation of statistics**

Idea: Let's mechanize these derivations

Goal: declarative specification $\longrightarrow$ executable code

$\Rightarrow$ **need a symbolic representation of statistics**

Our approach

1. Rigorously defined mathematical language
    - enables stating statistical problems
2. *Schemas* – program transformations
    - embodiments of mathematical reformulations
3. Interactive algorithm assistant
    - enables exploring the space of correct solutions

Challenge 1: Statistics is big

Many types of mathematics

- probability, optimization, calculus, linear algebra, ...

Challenge 1: Statistics is big

Many types of mathematics

- probability, optimization, calculus, linear algebra, ...

Computational aspects

- algorithms & datastructures
- numerical stability, robustness
- programming expertise and code tuning

Challenge 1: Statistics is big

Many types of mathematics

- probability, optimization, calculus, linear algebra, ...

Computational aspects

- algorithms & datastructures
- numerical stability, robustness
- programming expertise and code tuning

Myriad solution strategies

- e.g., SVMs, EM, L2E, NMF, various optimizers, ...
- domain-driven customizations

Challenge 1: Statistics is big

Many types of mathematics

- probability, optimization, calculus, linear algebra, ...

Computational aspects

- algorithms & datastructures
- numerical stability, robustness
- programming expertise and code tuning

Myriad solution strategies

- e.g., SVMs, EM, L2E, NMF, various optimizers, ...
- domain-driven customizations

$\Longrightarrow$ Need an *expressive* symbolic representation of mathematics.

Challenge 2: Ensuring correctness

Expressive representation $\longrightarrow$ easier to create nonsensical expressions

- especially true for mechanically generated expressions

Challenge 2: Ensuring correctness

Expressive representation $\longrightarrow$ easier to create nonsensical expressions

- especially true for mechanically generated expressions

Many schemas $\longrightarrow$ more chances for errors to sneak in

Challenge 2: Ensuring correctness

Expressive representation $\longrightarrow$ easier to create nonsensical expressions

- especially true for mechanically generated expressions

Many schemas $\longrightarrow$ more chances for errors to sneak in

$\Longrightarrow$ We use type theory to promote correctness.

Type Theory 101

The usual story: Types are a way to rule out ill-formed expressions.

Type Theory 101

The usual story: Types are a way to rule out ill-formed expressions.

The bigger story: Type theory connects mathematics and computation.

Type Theory 101

The usual story: Types are a way to rule out ill-formed expressions.

The bigger story: Type theory connects mathematics and computation.

1. Advanced type theories can be used to formalize mathematics.
    - an alternative to set theory
    - forms the basis of several modern theorem provers

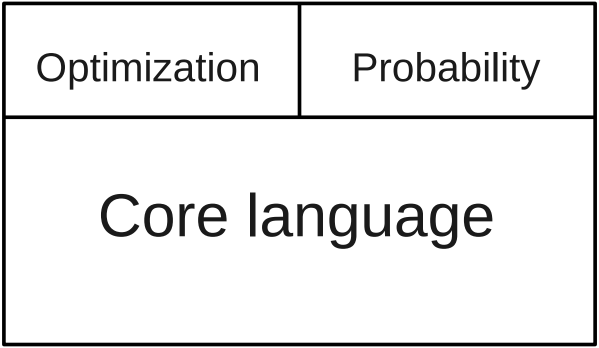
2. These type theories have a computational interpretation.
    - Curry-Howard: type systems are logics
    - increased mathematical precision about programs

Language

language = syntax + type system + semantics

Language

language = syntax + type system + semantics



Language: Core

$$T ::= \texttt{Bool} \mid \texttt{Int} \mid \texttt{Real} \mid T_1 \times \ldots \times T_n \mid T_1 \to T_2$$

$$\begin{array}{llr} E ::= & \texttt{true} \mid \texttt{false} \mid \neg E \mid E_1 \vee E_2 \mid E_1 \wedge E_2 & \text{Booleans} \\ & \mid r \mid E_1 + E_2 \mid E_1 * E_2 \mid E_1^{E_2} \mid \log E & \text{reals} \\ & \mid (E_1, \ldots, E_n) \mid E.k & \text{tuples} \\ & \mid \texttt{if}\ E_1\ \texttt{then}\ E_2\ \texttt{else}\ E_3 \mid E_1 = E_2 \mid E_1 \leq E_2 & \text{conditions} \\ & \mid x \mid \lambda x : T \,.\, E \mid E_1\ E_2 \mid \texttt{fix}\ E & \text{functional core} \end{array}$$

- A typed lambda calculus + recursion.

Language: Optimization

$$E \mathrel{+}= \operatorname*{arg\,max}_{x_1 : T_1, \ldots, x_n : T_n} \{E_1 \mid E_2\} \qquad \text{optimization}$$

- Full account: "Automating Mathematical Program Transformations" (Agarwal et al., PADL 2010).

Challenges in modeling continuous probability distributions

Continuous random variables arise naturally in numerous applications.

Challenges in modeling continuous probability distributions

Continuous random variables arise naturally in numerous applications.

- Almost all probabilistic languages do not support them.

Challenges in modeling continuous probability distributions

Continuous random variables arise naturally in numerous applications.

- Almost all probabilistic languages do not support them.

Why?

Challenges in modeling continuous probability distributions

Continuous random variables arise naturally in numerous applications.

- Almost all probabilistic languages do not support them.

Why?

- Computation and semantics are not as straightforward anymore.
  - weighted lists & summation vs. integration
  - We use symbolic reasoning.

Challenges in modeling continuous probability distributions

Continuous random variables arise naturally in numerous applications.

- Almost all probabilistic languages do not support them.

Why?

- Computation and semantics are not as straightforward anymore.
  - weighted lists & summation vs. integration
  - We use symbolic reasoning.

- Probability density functions now require more attention.
  - Mixed discrete-continuous densities require extra bookkeeping.
  - Not all distributions have a density.
  - We implement the necessary bookkeeping and restrictions.

Challenges in modeling continuous probability distributions

Continuous random variables arise naturally in numerous applications.

- Almost all probabilistic languages do not support them.

Why?

- Computation and semantics are not as straightforward anymore.
  - weighted lists & summation vs. integration
  - We use symbolic reasoning.
- Probability density functions now require more attention.
  - Mixed discrete-continuous densities require extra bookkeeping.
  - Not all distributions have a density.
  - We implement the necessary bookkeeping and restrictions.
- Functions must be measurable.
  - Non-measurable functions are not constructible in our language.

Language: Probability – new!

$$T \mathrel{+}= \texttt{Prob}\ T$$

$$\begin{array}{lr}
E \mathrel{+}= \texttt{bernoulli}\ E \mid \texttt{normal}\ E_1\ E_2 \mid \ldots & \text{distributions} \\
\quad \mid \texttt{return}\ E & \text{singleton distribution} \\
\quad \mid \texttt{var}\ x \sim E_1\ \texttt{in}\ E_2 & \text{random variables} \\
\quad \mid \texttt{condition}\ E_1\ \texttt{in}\ E_2 & \text{conditional distributions} \\
\quad \mid \mathbb{E}_{x \sim E_1}(E_2) & \text{expectation} \\
\quad \mid \texttt{pdf}\ E & \text{density functions}
\end{array}$$

- Probability monad (Giry 1981, Ramsey & Pfeffer 2003).
- **Continuous random variables!** (e.g., `Prob Real`)
- Semantics in terms of measure theory.

Modeling example revisited

Recall the example
from before:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\arg\max_{\theta} \; f(x \mid \theta)$$

Modeling example revisited

Recall the example from before:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\arg\max_{\theta} \; f(x \mid \theta)$$

In our language:

$$
\begin{array}{l}
\texttt{let}\ F\ \theta = \\
\quad \texttt{var}\ X_1 \ \sim\ \texttt{normal}\ \theta\ 1\ \texttt{in} \\
\quad \texttt{var}\ X_2 \ \sim\ \texttt{normal}\ \theta\ 1\ \texttt{in} \\
\quad \texttt{var}\ X_3 \ \sim\ \texttt{normal}\ \theta\ 1\ \texttt{in} \\
\quad\quad \texttt{return}\ (X_1, X_2, X_3) \\
\texttt{in} \\
\quad \arg\max_{\theta:\texttt{Real}} \{\ \texttt{pdf}\ (F\ \theta)\ (x_1, x_2, x_3) \mid \texttt{true}\ \}
\end{array}
$$

Modeling example revisited

Recall the example from before:

$$X_i \sim \text{Normal}(\theta, 1)$$
$$\arg\max_{\theta} \; f(x \mid \theta)$$

In our language:

$$
\begin{array}{l}
\texttt{let } F\ \theta = \\
\quad \texttt{var } X_1 \ \sim\ \texttt{normal}\ \theta\ 1\ \texttt{in} \\
\quad \texttt{var } X_2 \ \sim\ \texttt{normal}\ \theta\ 1\ \texttt{in} \\
\quad \texttt{var } X_3 \ \sim\ \texttt{normal}\ \theta\ 1\ \texttt{in} \\
\qquad \texttt{return } (X_1, X_2, X_3) \\
\texttt{in} \\
\quad \arg\max_{\theta:\texttt{Real}} \{\ \texttt{pdf}\ (F\ \theta)\ (x_1, x_2, x_3) \mid \texttt{true}\ \}
\end{array}
$$

- Mechanization requires a higher level of formalism.

Schemas

Logically: expression reformulation theorems

- example: $\forall a, b \in \mathbb{R},\ a * b = 0 \ \mapsto \ a = 0 \vee b = 0$

Schemas

Logically: expression reformulation theorems

- example: $\forall a, b \in \mathbb{R},\ a * b = 0 \ \mapsto\ a = 0 \vee b = 0$

Operationally: rewrite rules (implemented as OCaml functions)

Schemas

Logically: expression reformulation theorems

- example: $\forall a, b \in \mathbb{R},\ a * b = 0 \ \mapsto\ a = 0 \vee b = 0$

Operationally: rewrite rules (implemented as OCaml functions)

Current schema library contains 100+ schemas

- computer algebra
- propositional logic
- equation manipulation
- calculus
- optimization
- probability & statistics

The *Expectation-Maximization (EM)* schema

EM is widely used for maximum likelihood estimation (MLE).

- will require everything introduced so far

The *Expectation-Maximization (EM)* schema

EM is widely used for maximum likelihood estimation (MLE).

- will require everything introduced so far

$$\operatorname*{arg\,max}_{\theta:\texttt{Real}} \ \texttt{pdf}\left(\qquad\qquad\qquad \right) x \quad \mapsto$$

The *Expectation-Maximization (EM)* schema

EM is widely used for maximum likelihood estimation (MLE).

- will require everything introduced so far

$$\operatorname*{arg\,max}_{\theta:\texttt{Real}} \ \texttt{pdf} \left(\begin{array}{c} \texttt{var}\ Z \sim F_Z\ \texttt{in} \\ \texttt{var}\ X \sim F_X\ \texttt{in} \\ \texttt{return}\ X \end{array} \right) x \quad \mapsto$$

The *Expectation-Maximization (EM)* schema

EM is widely used for maximum likelihood estimation (MLE).

- will require everything introduced so far

$$\operatorname*{arg\,max}_{\theta:\mathtt{Real}} \ \mathtt{pdf} \begin{pmatrix} \mathtt{var}\ Z \sim F_Z\ \mathtt{in} \\ \mathtt{var}\ X \sim F_X\ \mathtt{in} \\ \mathtt{return}\ X \end{pmatrix} x \quad \mapsto$$

$$\begin{array}{l} \mathtt{let\ rec\ loop}\ \hat{\theta} = \\ \quad \mathtt{if}\ (\ldots)\ \mathtt{then}\ \hat{\theta} \\ \quad \mathtt{else\ loop}\ \operatorname{arg\,max}_{\theta:\mathtt{Real}} \mathbb{E}_{z\sim C}(\log(\mathtt{pdf}\ J\ (x, z))) \\ \mathtt{in} \\ \quad \mathtt{loop}\ \theta_0 \end{array}$$

The *Expectation-Maximization (EM)* schema

EM is widely used for maximum likelihood estimation (MLE).

- will require everything introduced so far

$$\arg\max_{\theta:\texttt{Real}} \ \texttt{pdf} \left(\begin{array}{c} \texttt{var } Z \sim F_Z \texttt{ in} \\ \texttt{var } X \sim F_X \texttt{ in} \\ \texttt{return } X \end{array} \right) x \quad \mapsto$$

$$\begin{array}{l} \texttt{let rec loop } \hat{\theta} = \\ \quad \texttt{if } (\ldots) \texttt{ then } \hat{\theta} \\ \quad \texttt{else loop } \arg\max_{\theta:\texttt{Real}} \mathbb{E}_{z \sim C}(\log(\texttt{pdf } J \ (x, z))) \\ \texttt{in} \\ \quad \texttt{loop } \theta_0 \end{array}$$

$$J = \left(\begin{array}{l} \texttt{var } Z \sim F_Z \texttt{ in} \\ \texttt{var } X \sim F_X \texttt{ in} \\ \quad \texttt{return } (X, Z) \end{array} \right) \quad C' = \left(\begin{array}{l} \texttt{var } Z \sim F_Z \texttt{ in} \\ \texttt{var } X \sim F_X \texttt{ in} \\ \texttt{condition } X = x \texttt{ in} \\ \quad \texttt{return } Z \end{array} \right) \quad C = C'[\theta := \hat{\theta}]$$

The *Expectation-Maximization (EM)* schema

EM is widely used for maximum likelihood estimation (MLE).

- will require everything introduced so far

$$\arg\max_{\theta:\texttt{Real}} \ \texttt{pdf} \begin{pmatrix} \texttt{var}\ Z \sim F_Z\ \texttt{in} \\ \texttt{var}\ X \sim F_X\ \texttt{in} \\ \texttt{return}\ X \end{pmatrix} x \quad \mapsto$$

$$\begin{array}{l} \texttt{let rec loop}\ \hat{\theta} = \\ \quad \texttt{if}\ (\ldots)\ \texttt{then}\ \hat{\theta} \\ \quad \texttt{else loop}\ \arg\max_{\theta:\texttt{Real}} \mathbb{E}_{z\sim C}(\log(\texttt{pdf}\ J\ (x,z))) \\ \texttt{in} \\ \quad \texttt{loop}\ \theta_0 \end{array}$$

$$J = \begin{pmatrix} \texttt{var}\ Z \sim F_Z\ \texttt{in} \\ \texttt{var}\ X \sim F_X\ \texttt{in} \\ \texttt{return}\ (X, Z) \end{pmatrix} \quad C' = \begin{pmatrix} \texttt{var}\ Z \sim F_Z\ \texttt{in} \\ \texttt{var}\ X \sim F_X\ \texttt{in} \\ \texttt{condition}\ X = x\ \texttt{in} \\ \texttt{return}\ Z \end{pmatrix} \quad C = C'[\theta := \hat{\theta}]$$

Relies crucially on symbolic operations.

Interactive algorithm assistant

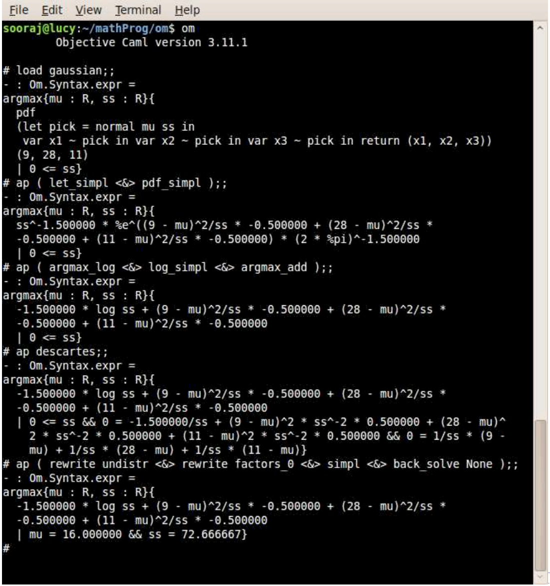
Features

- enter problems
- apply schemas
- undo/redo
- combinators

Status

- can solve several textbook examples of MLE, incl. via EM
- autotuning + more sophisticated code generation is planned

Come see me for a demo!



Conclusions

- The first symbolic formalization of statistics for freely expressing statistical problems and reformulations
  - type-theoretic formalization of probability & optimization
  - continuous probability distributions
- An implementation of the language & schemas
  - the Expectation-Maximization (EM) schema
  - interactive algorithm assistant
- Future plans include incorporating feedback
  - autotuning, high-performance code generation
  - model selection, causal inference

Thank you

Fin.