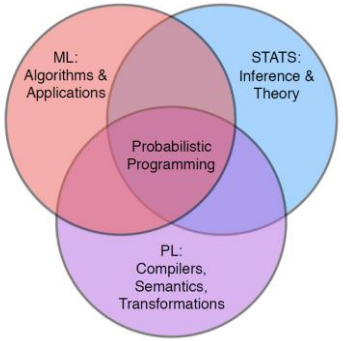
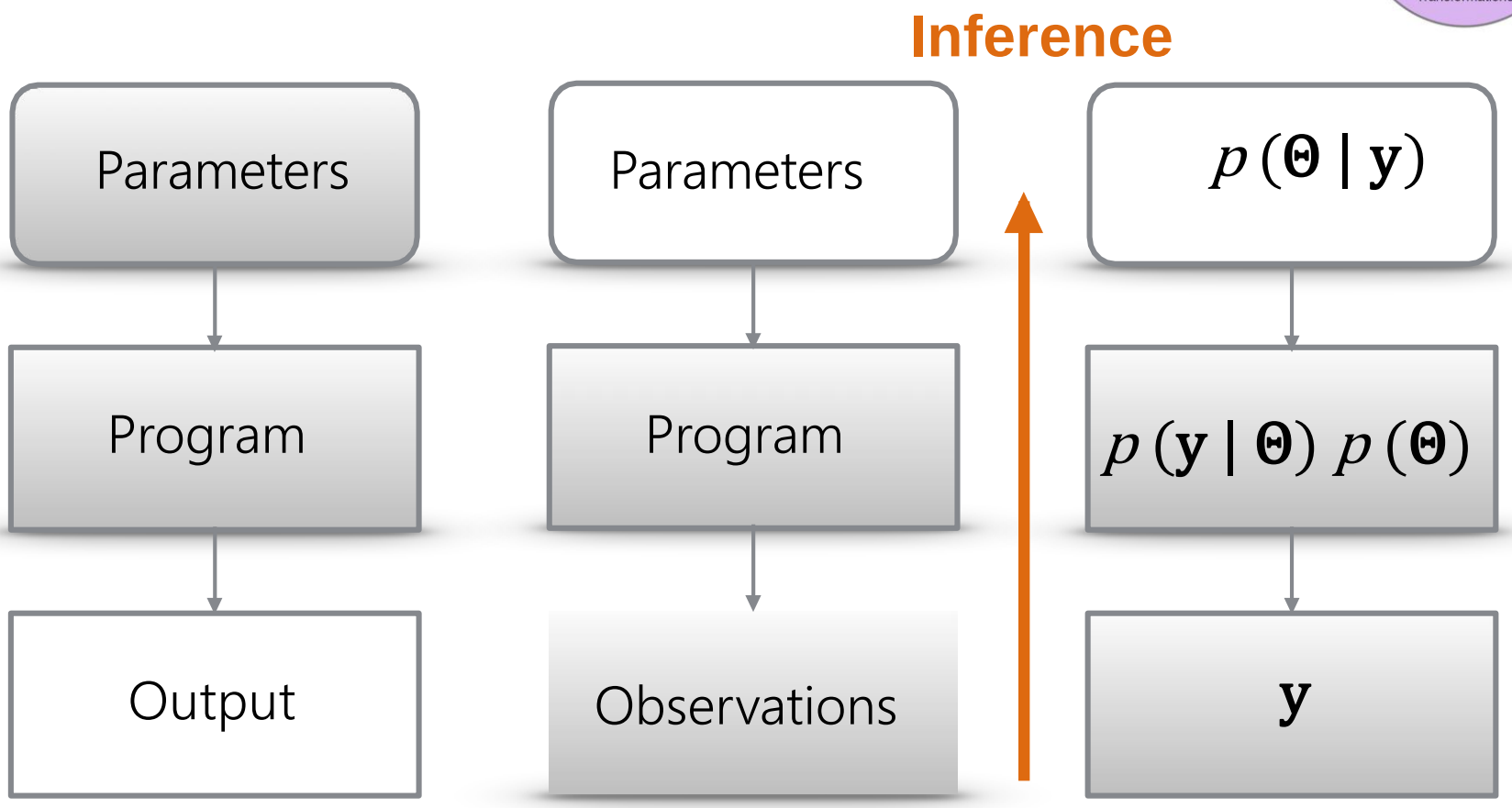
Probabilistic & Approximate Computing

Sasa Misailovic

UIUC

Intuition





Programming

Probabilistic Programming

Statistics

** Slide from Frank Wood*

Example: Bernoulli Program

(| X | Y |
|---|---|
| -- | -- |
, 1.0)

```
X := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | -- |
, 0.7), (| X | Y |
|---|---|
| False | -- |
, 0.3)

```
Y := not X;
```

(| X | Y |
|---|---|
| True | False |
, 0.7), (| X | Y |
|---|---|
| False | True |
, 0.3)

Example: Bernoulli Program

(| X | Y |
|---|---|
| -- | -- |
, 1.0)

```
X := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | -- |
, 0.7), (| X | Y |
|---|---|
| False | -- |
, 0.3)

```
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |
, 0.49), (| X | Y |
|---|---|
| True | False |
, 0.21), (| X | Y |
|---|---|
| False | True |
, 0.21), (| X | Y |
|---|---|
| False | False |
, 0.09)

Example: Bernoulli Program

(| X | Y |
|---|---|
| -- | -- |
| | | , 1.0)

```
X := Bernoulli(0.7);
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |
| | | , 0.49), (| X | Y |
|---|---|
| True | False |
| | | , 0.21), (| X | Y |
|---|---|
| False | True |
| | | , 0.21), (| X | Y |
|---|---|
| False | False |
| | | , 0.09)

```
condition ( X == True );
return Y;
```

(| X | Y |
|---|---|
| True | True |
| | | , 0.49/0.7), (| X | Y |
|---|---|
| True | False |
| | | , 0.21/0.7),

Example: Bernoulli Program

(| X | Y |
|---|---|
| -- | -- |

, 1.0)

```
X := Bernoulli(0.7);
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.21), (| X | Y |
|---|---|
| False | False |

, 0.09)

```
condition ( X == Y );
return Y;
```

(| X | Y |
|---|---|
| True | True |

, 0.49/0.58), (| X | Y |
|---|---|
| False | False |

, 0.09/0.58)

Example: Bernoulli Program

(| X | Y |
|---|---|
| -- | -- |

, 1.0)

```
X := Bernoulli(0.7);
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.21), (| X | Y |
|---|---|
| False | False |

, 0.09)

```
factor ( X ? 0 : -1 );
```

$e^0 = 1$

$e^{-1} = 0.37$

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.077)(| X | Y |
|---|---|
| False | False |

, 0.033)

Example: Bernoulli Program

factor => condition

(| X | Y |
|---|---|
| -- | -- |

, 1.0)

```
X := Bernoulli(0.7);
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.21), (| X | Y |
|---|---|
| False | False |

, 0.09)

```
factor ( X ? 0 : -∞ );
```

$e^0 = 1$

$e^{-\infty} = 0$

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0), (| X | Y |
|---|---|
| False | False |

, 0)

Example: Bernoulli Program

(| X | Y |
|---|---|
| -- | -- |

, 1.0)

```
X := Bernoulli(0.7);
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.21), (| X | Y |
|---|---|
| False | False |

, 0.09)

```
condition ( X == Y );
return Y;
```

(| X | Y |
|---|---|
| True | True |

, 0.49/0.58), (| X | Y |
|---|---|
| False | False |

, 0.09/0.58)

Example: Bernoulli Program

(| X | Y |
|---|---|
| -- | -- |

, 1.0)

```
X := Bernoulli(0.7);
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.21), (| X | Y |
|---|---|
| False | False |

, 0.09)

```
factor ( X ? 0 : -1 );
```

$e^0 = 1$

$e^{-1} = 0.37$

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.077)(| X | Y |
|---|---|
| False | False |

, 0.033)

Example: Bernoulli Program

factor => condition

(| X | Y |
|---|---|
| -- | -- |

, 1.0)

```
X := Bernoulli(0.7);
Y := Bernoulli(0.7);
```

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0.21), (| X | Y |
|---|---|
| False | False |

, 0.09)

```
factor ( X ? 0 : -∞ );
```

$e^0 = 1$

$e^{-\infty} = 0$

(| X | Y |
|---|---|
| True | True |

, 0.49), (| X | Y |
|---|---|
| True | False |

, 0.21), (| X | Y |
|---|---|
| False | True |

, 0), (| X | Y |
|---|---|
| False | False |

, 0)

Handling Conditionals

```
if (b)
   S1
else
   S2
```

$\approx$

$p(X,Y,Z)$

```
observe b;
S1;
```

$p_1(X,Y,Z) \cdot [b]$

$p(X,Y,Z)$

```
observe !b;
S2;
```

$p_2(X,Y,Z) \cdot [!\,b]$

$$p_1(X,Y,Z) \cdot [b] + p_2(X,Y,Z) \cdot [!\,b]$$

(+ bookkeeping)

Handling Loops

```
N = Random(...)
for i in [0..N)
   S
```

≈

$p(X, Y, Z)$

```
N = Random(...)
assert (N <= Nmax);
for i in [0 .. Nmax) {
  if (i < N)
     S
}
```

Probability of state: $p_{loop}(X, Y, Z) \cdot p_{N \leq N_{max}}$

Probability of error: $1 - p_{N \leq N_{max}}$

Unrolling factor depends on property to check

e.g. `assert (X > 0) withPr > 0.99`

Beyond Bayesian Net Models

Geometric Distribution: Probability of the <u>number</u> of Bernoulli trials to get one success

```
var geometric = function() {
  return flip(.5) ? 0 : geometric() + 1;
}

var dist = Infer({method: 'enumerate', maxExecutions: 10},
                                 geometric);
viz.auto(dist);
```

Exact Inference

Naïve approach: Compute $P(x_1, x_2, \ldots, x_n)$

Better approach:

Take advantage of (conditional) independencies

- Whenever we can expose conditional independence, e.g., $\mathrm{P}_{X_1 X_2 | X_3}(x_1, \mathrm{x}_2 | \mathrm{x}_3) = \mathrm{P}_{X_1 | X_3}(\mathrm{x}_1 | \mathrm{x}_3) \cdot \mathrm{P}_{X_2 | X_3}(\mathrm{x}_2 | \mathrm{x}_3)$ the computation is more efficient
- Instead of the full table, keep smaller tables (for the individual variable distributions

Compute distributions from parents to children

Complexity of Exact Inference

Number of variables: $\boldsymbol{n}$

Naïve enumeration: complexity is $O(2^n)$

Variable Elimination: if the maximum number of parents of the nodes is $k \in \{1, \ldots, n\}$, then the complexity is $n \cdot O(2^k)$.

For many models this is a good improvement, but always possible to construct pathological models.

Exact inference: Pragmatics

- Use dynamic programming where possible

- The algorithms for inference in Bayesian Networks are directly applicable if the program can be translated to one

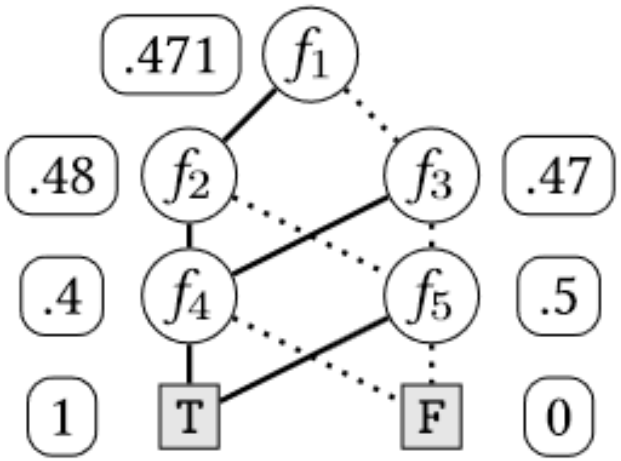
- Leverage techniques from model checking
  see ***model counting:*** compute the number of models (distinct variable assignments) for which some formula evaluates to true

Example of Model Checking

http://dicelang.cs.ucla.edu

```
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.

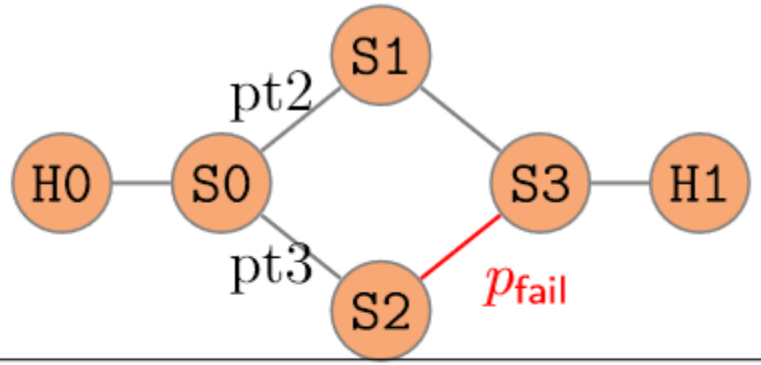


(b) Compiled BDD with weighted model counts.

Fig. 1. Illustration of compiling a Dice program that exploits factorization.

Scaling Exact Inference for Discrete Probabilistic Programs, OOPSLA 2020

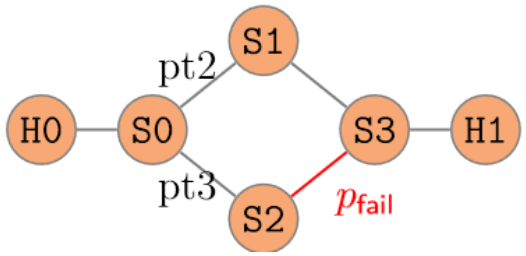
Example: Network Reliability *Bayonet, PLDI 2018*




```
topology{
  nodes{ H0, H1, S0, S1, S2, S3 }
  links{
    (H0,pt1) <-> (S0,pt1),
    (S0,pt2) <-> (S1,pt1),
    (S0,pt3) <-> (S2,pt1),
    (S1,pt2) <-> (S3,pt1),
    (S2,pt2) <-> (S3,pt2),
    (S3,pt3) <-> (H1,pt1)
  }
}
```

To try out:
https://dl.acm.org/do/10.1145/3211997/full/

Example: Network Reliability

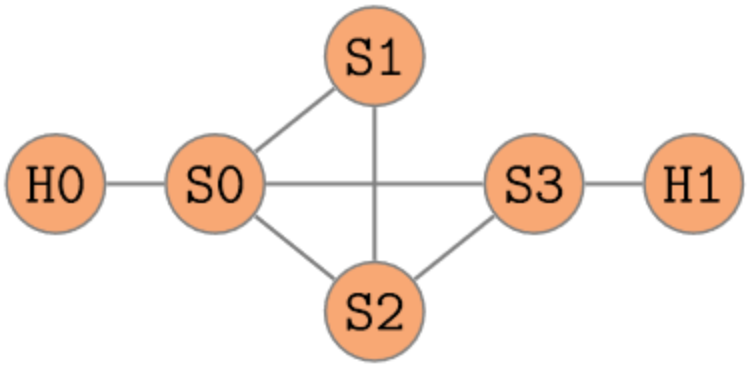



```
def h0(pkt, port){
  fwd(1);
}
def h1(pkt, port)
  state arrived(0)
{
  arrived=1;
  drop;
}
def s0(pkt, port){
  if flip(1/2){
    fwd(2);
  }else{
    fwd(3);
  }
}
```

```
def s1(pkt, port){
  fwd(2);
}
def s2(pkt, port)
  state fail(
    flip(1/1000))
{
  if fail {
    drop;
  } else {
    fwd(2);
  }
}
def s3(pkt, port){
  fwd(3);
}
```

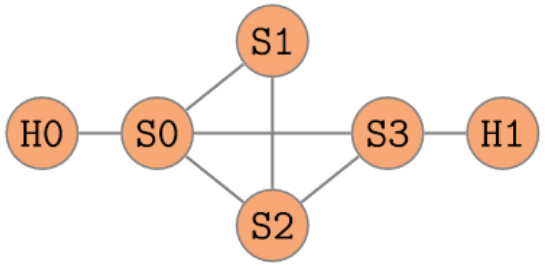
```
query probability(arrived@H1); // 0.9995
```

Example: Network Congestion




```
topology{
  nodes{ H0, H1, S0, S1, S2, S3
       }
  links{
    (H0,pt1) <-> (S0,pt1),
    (S0,pt2) <-> (S1,pt1),
    (S0,pt3) <-> (S3,pt1),
    (S0,pt4) <-> (S2,pt1),
    (S1,pt2) <-> (S2,pt2),
    (S2,pt3) <-> (S3,pt2),
    (S3,pt3) <-> (H1,pt1)
  }
}
queue_capacity 2;
```

Example: Network Congestion




```
def h0(pkt, port) state pkt_count(0){
  if pkt_count < 3 {
    new; pkt_count = pkt_count + 1; fwd(1);
  } else { drop; }
}
def s0(pkt, port){
  if flip(1/3){ fwd(1); }
  else if flip(1/2){ fwd(2); }
  else{ fwd(3); }
}
def s1(pkt, port){ fwd(2); }
def s2(pkt, port){ fwd(3); }
def s3(pkt, port){ fwd(3); }
def h1(pkt, port) state pkt_count(0){
  pkt_count = pkt_count + 1; drop;
}
query probability(pkt_count@H1 < 3); // ?
```

// $2663191003150725233/5997013881313296384 \approx 0.444$

Scheduler:

```
RunSw:=0, FwdQ:=1;

def scheduler(){
  actions := []: (ℝ × ℝ)[]; // list of actions
  for i in [0..k) { // k: number of nodes
    if (Q_in@i).size() > 0 {
      actions ~= [(RunSw, i)]; } // run node
    if (Q_out@i).size() > 0 {
      actions ~= [(FwdQ, i)]; // forward packet
    }
  }
  n := actions.length; // pick action u.a.r.:
  return actions[uniformInt(0,n-1)];
}
```

Continuous Models

The distributions in the program are continuous

- We are computing the log-likelihood of the trace

Doing ‘hard’ observations is ineffective: the probability of each observation is 0.

- Instead, use ‘factor’: the ‘soft’ version of conditioning

TrueSkill Model

Player Skill: Initially, we assume all players have a similar (randomly assigned) skill, centered around starting point

$$Skill_{Player} \sim Gaussian(100, 10)$$

Player Performance: it is based primarily on skill, but in every game it can be higher or lower, based on the player's inspiration

$$Performance_{Player} \sim Gaussian(Skill, 15)$$

Tournament Scores: Each player plays against others, we record the victory if the player's performance was higher than their opponent's

$$Performance_{Player\#1} > Performance_{Player\#2}$$

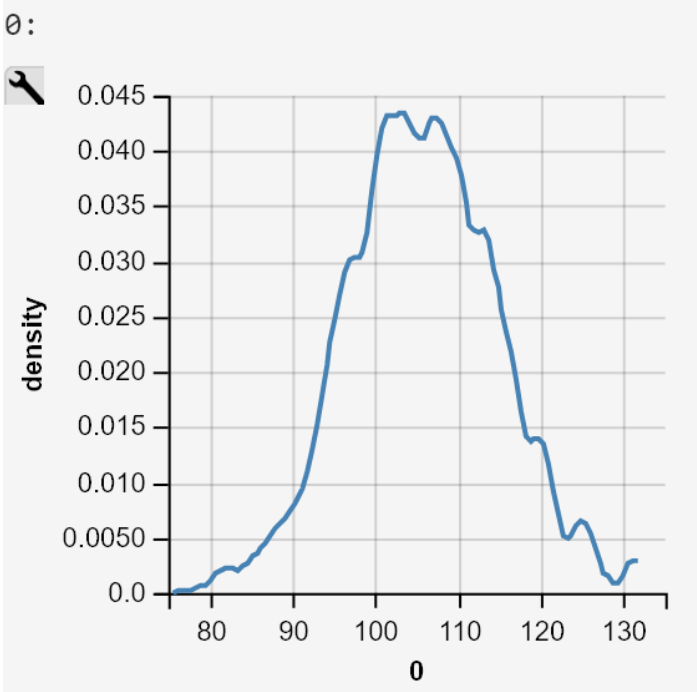
TrueSkill Example: 3 Players

```
var trueSkill = function(){

  var skillA = gaussian(100, 10);
  var skillB = gaussian(100, 10);
  var skillC = gaussian(100, 10);

  var perfA1 = gaussian(skillA, 15), perfB1 = gaussian(skillB, 15);
  condition (perfA1 > perfB1);

  var perfB2 = gaussian(skillB, 15), perfC2 = gaussian(skillC, 15);
  condition (perfB2 > perfC2);

  var perfA3 = gaussian(skillA, 15), perfC3 = gaussian(skillC, 15);
  condition (perfA3 > perfC3);

  return skillA;
}
```

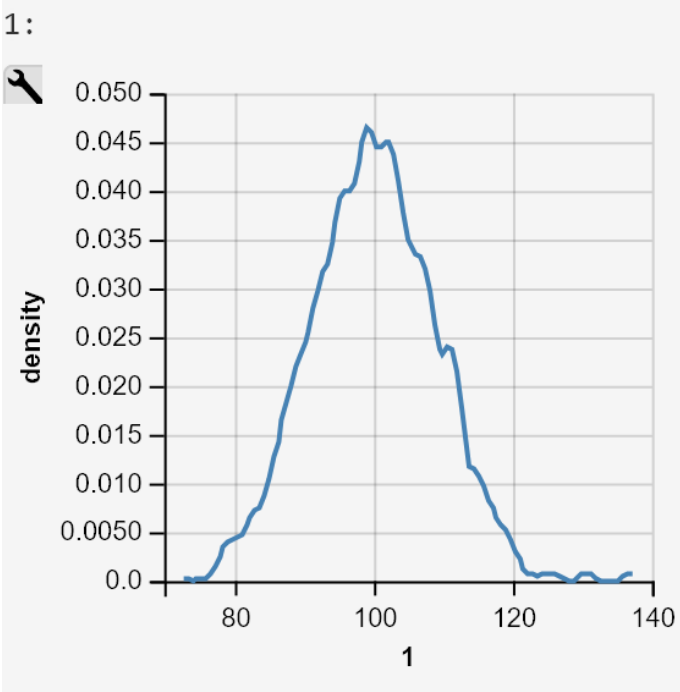
```
var res = Infer({method: 'MCMC', samples:
                 50000}, trueSkill)
print("Expected value: "+expectation(res));
viz.auto(res);
```

Performances

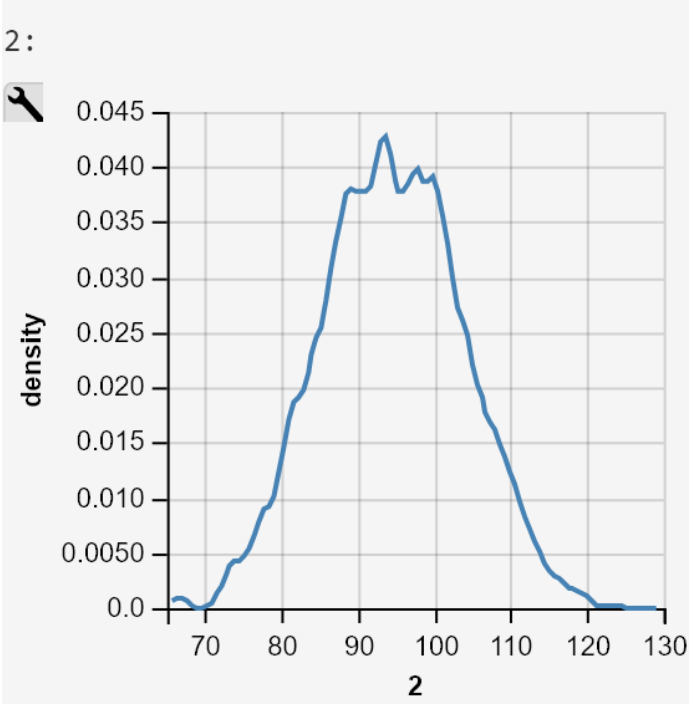
Player A



Player B



Player C



Mean: **105.3** **100.4** **94.6**

Try at Home

Extend the problem to n games:

- The results array contains playes and whether 1st won e.g.,

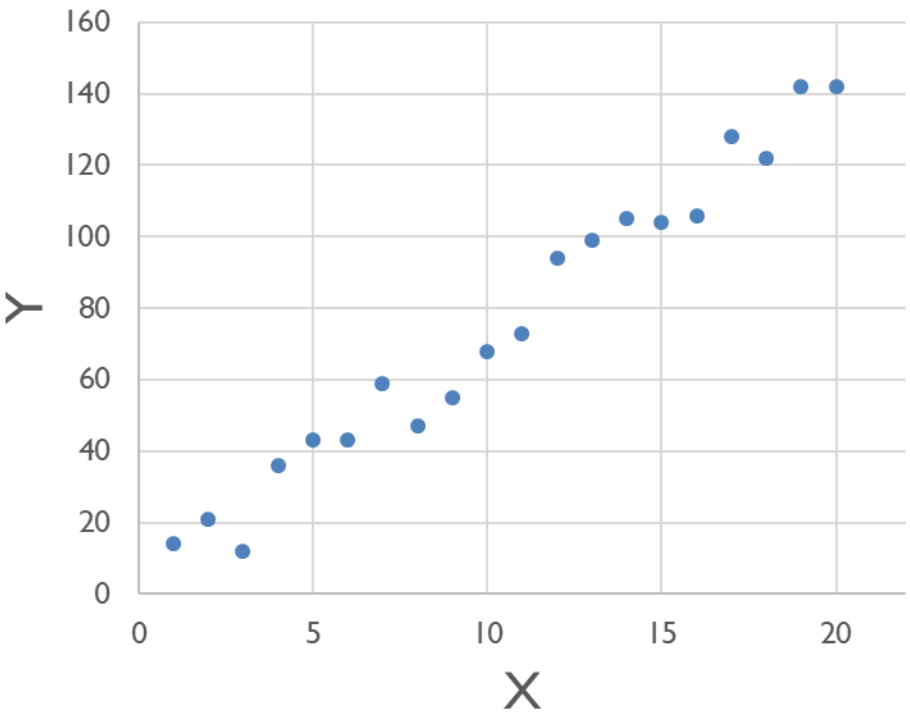
  Res = [['A', 'B', true], ['A', 'B', false], ...]
- Hint: define map M = {'A': skillA, 'B': skillB, 'C': skillC}
- Performance is now indexed by M[Res[0][0]] and M[Res[0][1]]
- Use map operator to condition on all elements of Res

Extend the problem to m players

- Use map operator to generate skills (may use number instead of letter for the player's name)

Continuous Models: Linear Regression

Given a set of points, find a linear relationship that most accurately describes this set



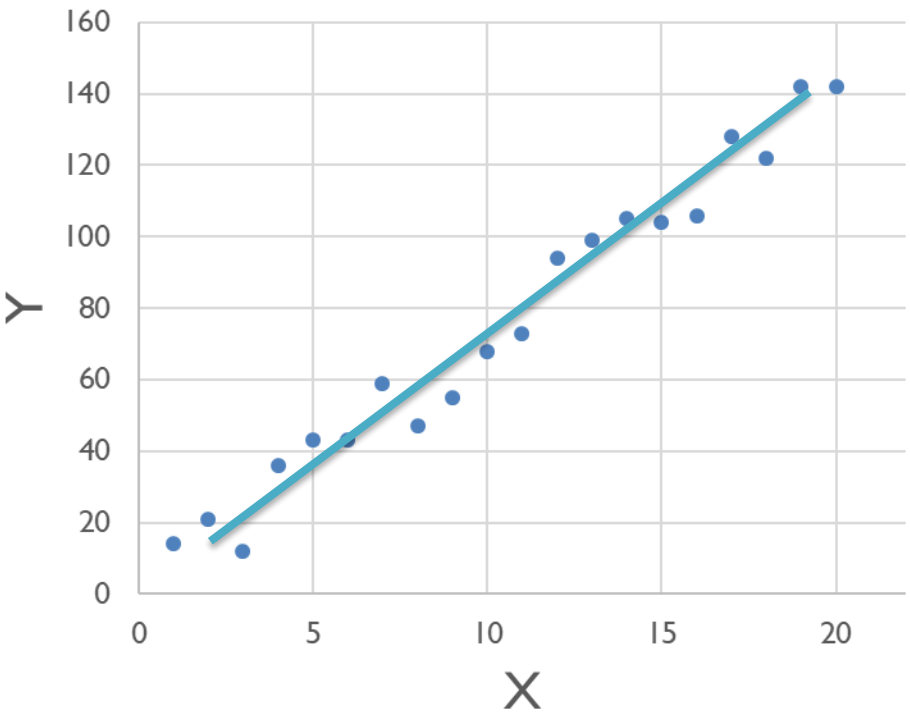
$$Y = \boldsymbol{w} \cdot X + \boldsymbol{b}$$

Slope

Intercept

Classical Solution (MLE)

Given a set of points, find a linear relationship that most accurately describes this set



$$Y = \boldsymbol{w} \cdot X + \boldsymbol{b}$$

Seeks point estimates (individual values) of w and b that minimize the error (e.g. square error) between the line and all the points

For the formulas and derivation see e.g.
https://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/06/lecture-06.pdf

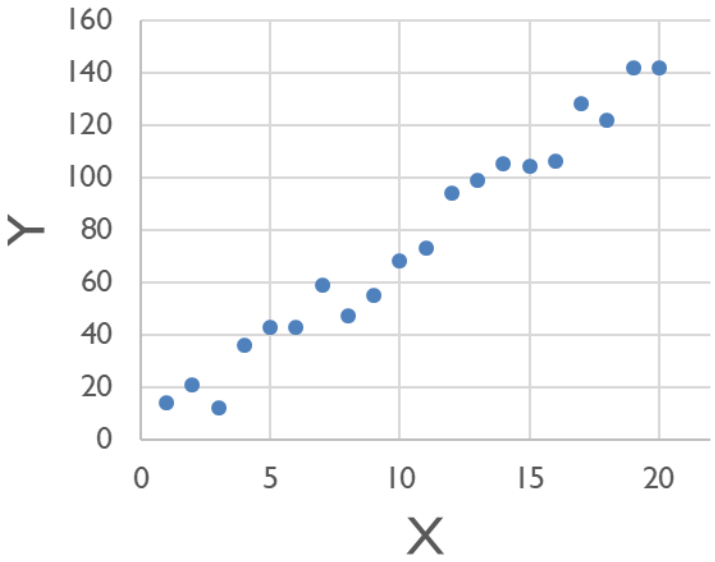
Linear Regression

```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Linear Regression

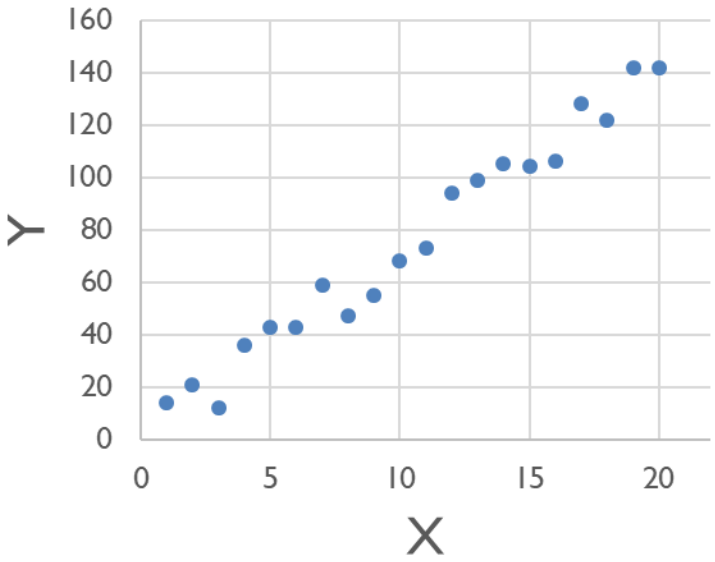
```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Linear Regression

```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

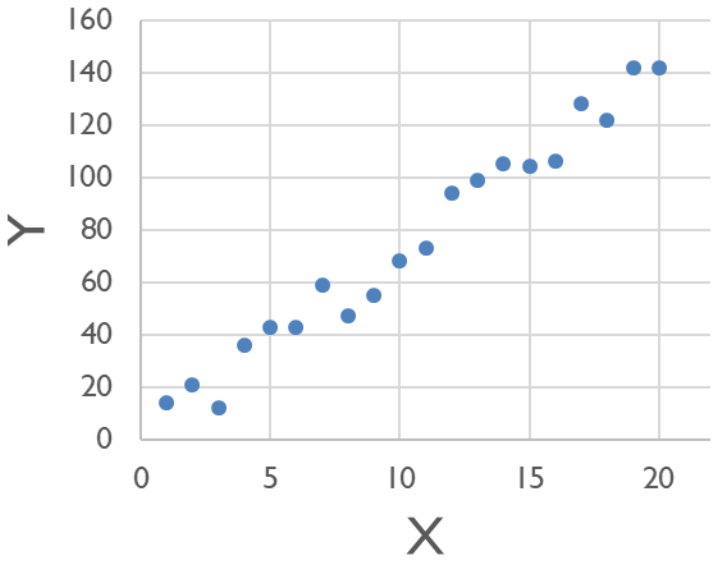
Datasets

```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

Conditioning on Data

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Linear Regression

```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

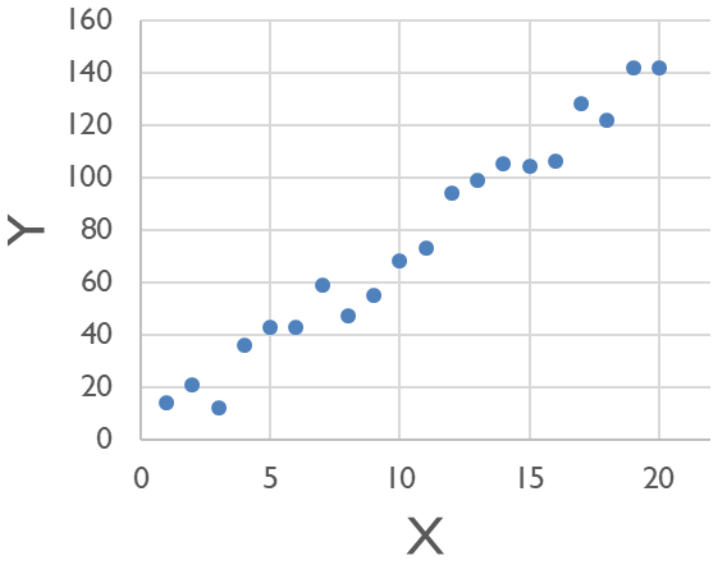
```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

Conditioning on Data

Queries

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Linear Regression

```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

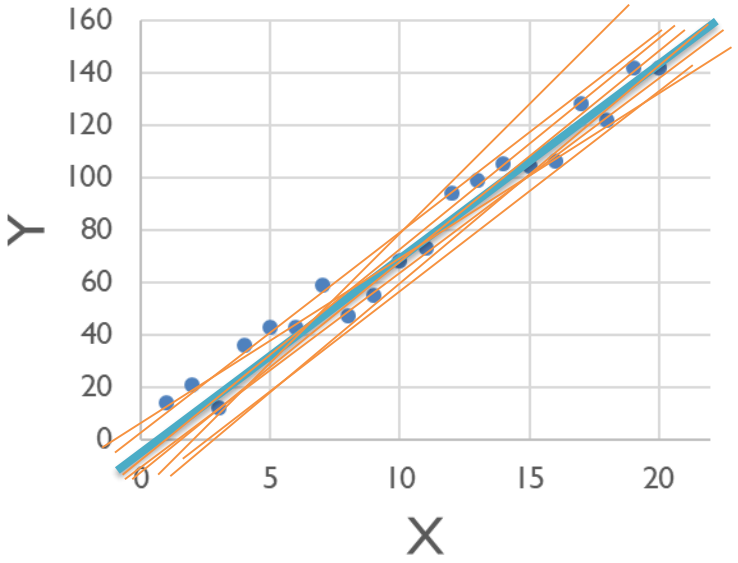
```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

Conditioning on Data

Queries

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Continuous Models: Linear Regression

```
var xs = [0, 1, 2, 3];  var ys = [0, 1, 4, 6];

var model = function() {
  var slope = gaussian(0, 2);
  var intercept = gaussian(0, 2);
  var sigma = 1; // for more interesting result, change to gamma(1, 1);

  var f = function(x) { return slope * x + intercept; };

  map2(
 // function(x, y) {  factor(Gaussian({mu: f(x), sigma: sigma}).score(y));  }, xs, ys);
    function(x, y) {  observe(Gaussian({mu: f(x), sigma: sigma}), y);  }, xs, ys); //same

  return [slope,intercept];
}

viz.marginals({method: 'MCMC', samples: 10000}, model));
```

Likelihood or Log-likelihood?

target = 0.0

```
X := Gaussian(0, 1);
```

target = $\mathcal{N}$_logpdf(X,0,1)

Likelihood or Log-likelihood?

target = 0.0

```
X := Gaussian(0, 1);
Y := Gaussian(X, 1);
```

target = $\mathcal{N}$_logpdf(X,0,1) + $\mathcal{N}$ _logpdf(Y,X,1)

Likelihood or Log-likelihood?

target = 0.0

```
X := Gaussian(0, 1);
Y := Gaussian(X, 1);
```

target = $\mathcal{N}$_logpdf(X,0,1) + $\mathcal{N}$ _logpdf(Y,X,1)

```
observe (Y, 0.5);
return X;
```

The probability that a continuous random variable has a particular value is equal to 0 (and logpdf(0) = -inf)

Likelihood or Log-likelihood?

target = 0.0

```
X := Gaussian(0, 1);
Y := Gaussian(X, 1);
```

target = $\mathcal{N}$_logpdf(X,0,1) + $\mathcal{N}$ _logpdf(Y,X,1)

```
observe (Y, 0.5);
```

```
factor (Gaussian.score(0.5, Y));
```

Likelihood or Log-likelihood?

target = 0.0

```
X := Gaussian(0, 1);
Y := Gaussian(X, 1);
```

target = $\mathcal{N}$_logpdf(X,0,1) + $\mathcal{N}$ _logpdf(Y,X,1)

```
factor (Gaussian.score(0.5, Y));
```

target = $\mathcal{N}$_logpdf(X,0,1) + $\mathcal{N}$_logpdf(Y,X,1)
+ $\mathcal{N}$_logpdf(0.5,X,1)

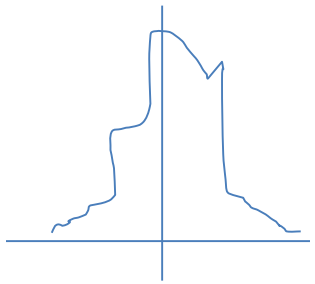
Approaches to Probabilistic Inference

Exact and Approximate

Sampling

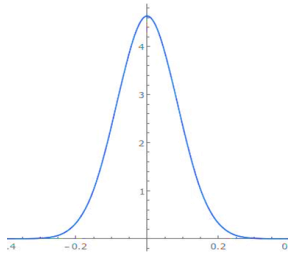
(Rejection – Church)

(MCMC – Church & Stan & R2)



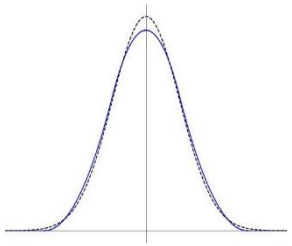
Variational Inference

(Fun & Infer.NET)

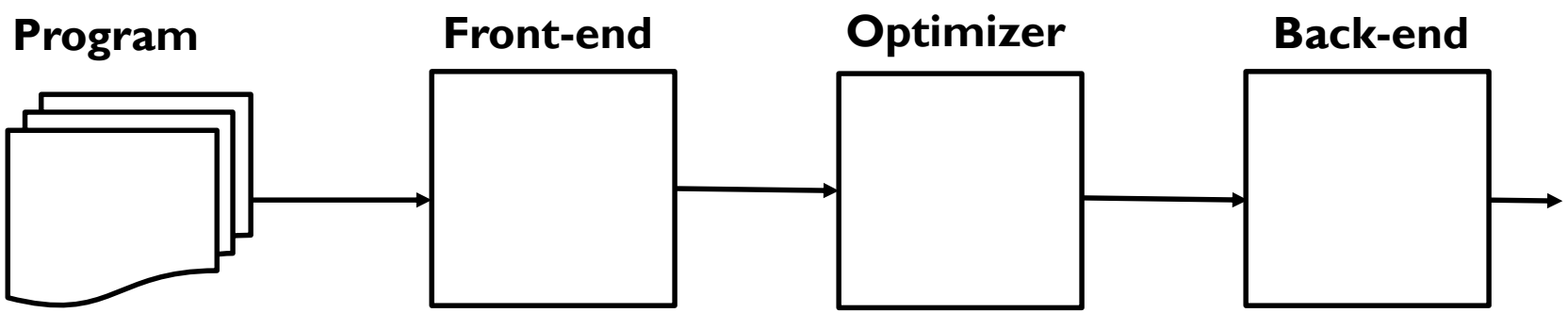


Exact Symbolic Inference

(PSI & Hakaru)

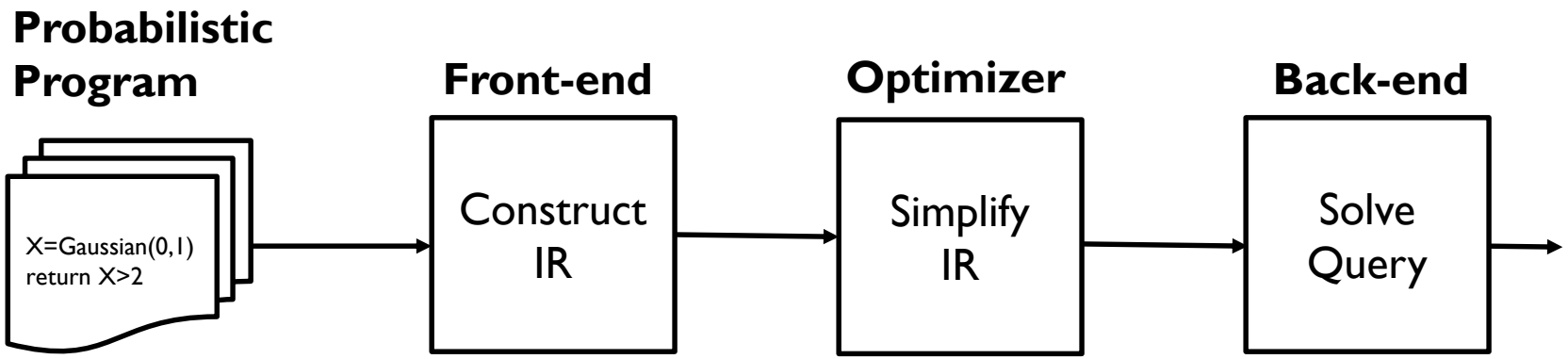


PSI Overview



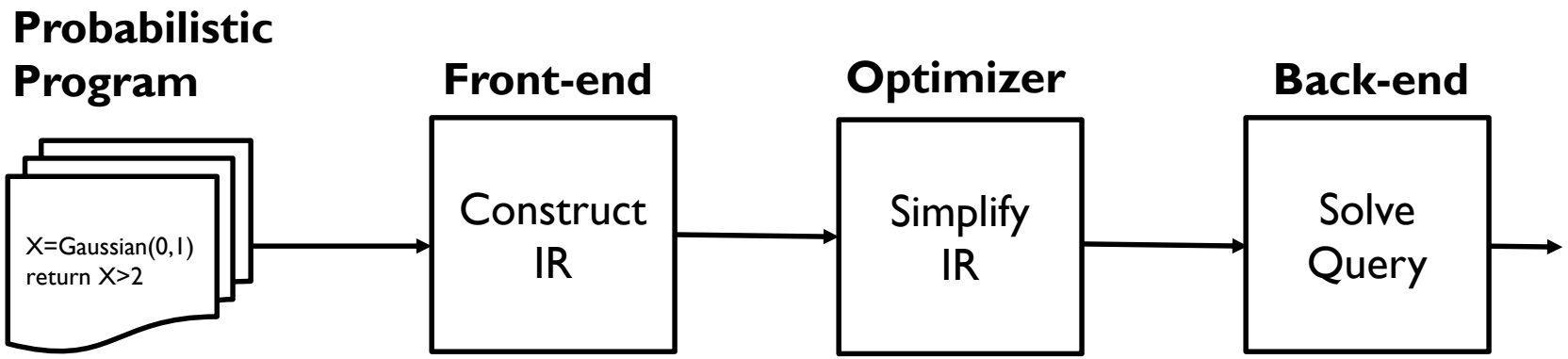
T. Gehr, S. Misailovic, M. Vechev; PSI: Exact Symbolic Inference for Probabilistic Programs (CAV 2016)

PSI Overview



$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2] \\
& \mid \int_{\mathbb{R}} \mathrm{d}x \, e[\![x]\!] \mid \sum_{x \in \mathbb{Z}} e[\![x]\!] \mid \varphi(e_1, \ldots, e_n)
\end{aligned}
$$

PSI Overview



- Basic algebraic simplifications

  $x + x \to 2 \cdot x, \quad x \cdot x \to x^2, \ldots$

- Simplifications on constraints

  $[x = 0] + [x \neq 0] \to 1, \quad [x \leq 0] \cdot [0 \leq x] \to [x = 0],$

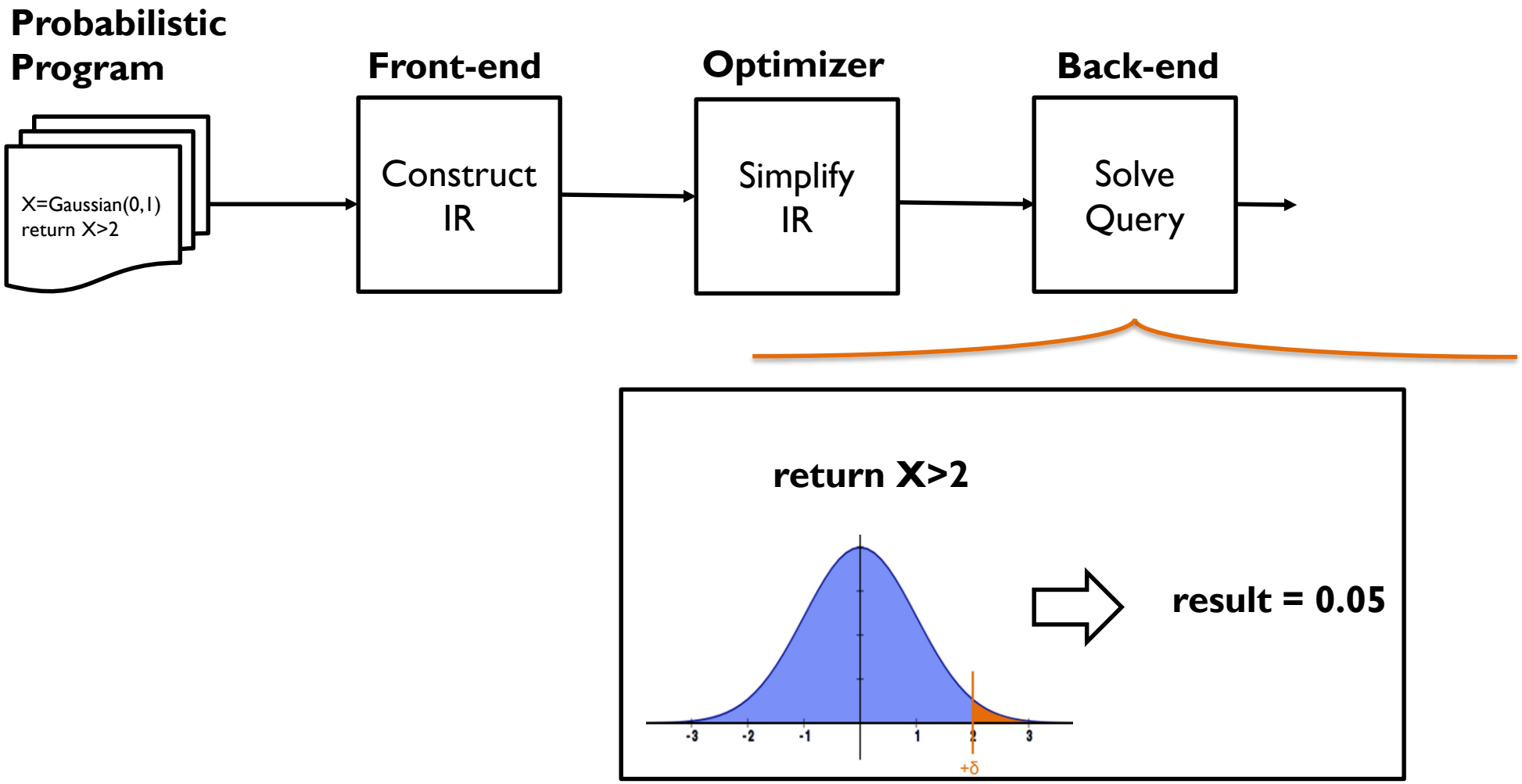
  $\delta(x) \cdot [1 \leq x] \to 0, \ldots$

- Symbolic Integration

  $\int_{\mathbb{R}} \mathrm{d}x \int_{\mathbb{R}} \mathrm{d}y\, [0 \leq x] \cdot [x \leq 1] \cdot [0 \leq y] \cdot [y \leq 1] \cdot \delta(z - x \cdot y) \to$

  $-[0 < z] \cdot [z \leq 1] \cdot \log(z), \ldots$

PSI Overview



PSI's Symbolic Domain

$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2]
\end{aligned}
$$

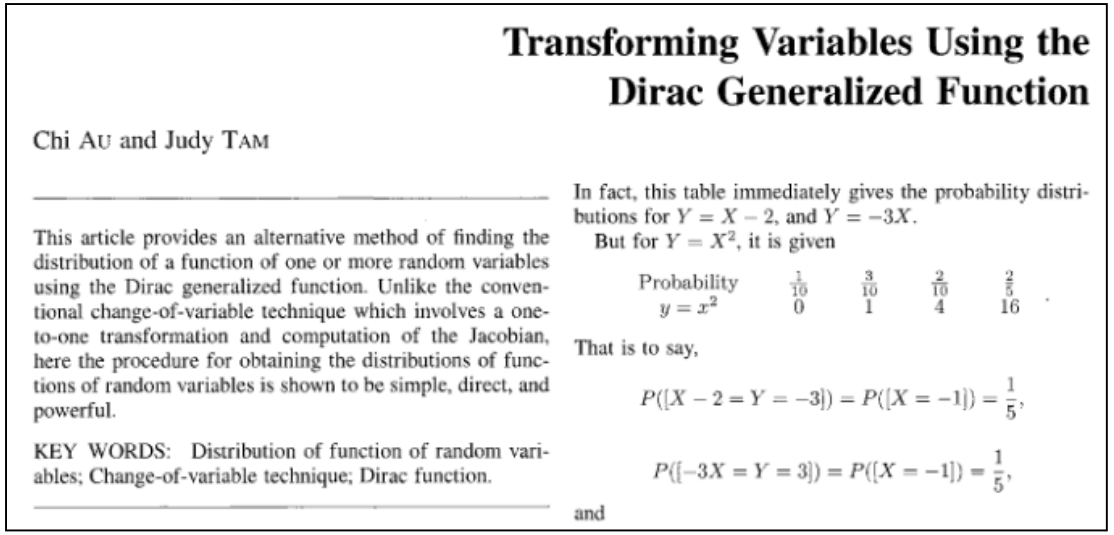
Encodes probability density functions

PSI's Symbolic Domain

$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2]
\end{aligned}
$$

$$
\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x)\, dx = 1
$$

Encodes probability density functions



Transforming Variables Using the Dirac Generalized Function

Chi Au and Judy Tam

This article provides an alternative method of finding the distribution of a function of one or more random variables using the Dirac generalized function. Unlike the conventional change-of-variable technique which involves a one-to-one transformation and computation of the Jacobian, here the procedure for obtaining the distributions of functions of random variables is shown to be simple, direct, and powerful.

KEY WORDS: Distribution of function of random variables; Change-of-variable technique; Dirac function.

In fact, this table immediately gives the probability distributions for $Y = X - 2$, and $Y = -3X$.

But for $Y = X^2$, it is given

Probability	$\frac{1}{10}$	$\frac{3}{10}$	$\frac{2}{10}$	$\frac{2}{5}$
$y = x^2$	0	1	4	16

That is to say,

$$P([X - 2 = Y = -3]) = P([X = -1]) = \frac{1}{5},$$

$$P([-3X = Y = 3]) = P([X = -1]) = \frac{1}{5},$$

and

PSI's Symbolic Domain

$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2]
\end{aligned}
$$

Encodes probability density functions

$$\mathsf{Bernoulli}(x; p) = p \cdot \delta(1 - x) + (1 - p) \cdot \delta(x)$$

PSI's Symbolic Domain

$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2]
\end{aligned}
$$

Encodes probability density functions

$$\mathsf{Bernoulli}(x; p) = p \cdot \delta(1 - x) + (1 - p) \cdot \delta(x)$$

$$\mathsf{Gauss}(x; \mu, \nu) = [\nu = 0] \cdot \delta(x - \mu) + [\nu \neq 0] \cdot \frac{\mathsf{e}^{-(x-\mu)^2/(2\nu)}}{(2\pi\nu)^{\frac{1}{2}}}$$

PSI's Symbolic Domain

$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2] \\
& \mid \int_{\mathbb{R}} \mathsf{d}x \, e[\![x]\!] \mid \sum_{x \in \mathbb{Z}} e[\![x]\!] \mid \varphi(e_1, \ldots, e_n)
\end{aligned}
$$

Encodes probability density functions

$$\mathsf{Bernoulli}(x; p) = p \cdot \delta(1 - x) + (1 - p) \cdot \delta(x)$$

$$\mathsf{Gauss}(x; \mu, \nu) = [\nu = 0] \cdot \delta(x - \mu) + [\nu \neq 0] \cdot \frac{\mathsf{e}^{-(x-\mu)^2/(2\nu)}}{(2\pi\nu)^{\frac{1}{2}}}$$

PSI's Symbolic Domain

$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2] \\
& \mid \int_{\mathbb{R}} \mathsf{d}x \, e[\![x]\!] \mid \sum_{x \in \mathbb{Z}} e[\![x]\!] \mid \varphi(e_1, \ldots, e_n)
\end{aligned}
$$

Encodes probability density functions

$$\mathsf{Bernoulli}(x; p) = p \cdot \delta(1 - x) + (1 - p) \cdot \delta(x)$$

$$\mathsf{Gauss}(x; \mu, \nu) = [\nu = 0] \cdot \delta(x - \mu) + [\nu \neq 0] \cdot \frac{\mathsf{e}^{-(x-\mu)^2/(2\nu)}}{(2\pi\nu)^{\frac{1}{2}}}$$

$$\mathsf{UniformInt}(x; a, b) = \frac{\sum_{x' \in \mathbb{Z}} \delta(x - x') \cdot [a \leq x'] \cdot [x' \leq b]}{\sum_{x' \in \mathbb{Z}} [a \leq x'] \cdot [x' \leq b]}$$

PSI's Symbolic Domain

$$
\begin{aligned}
e \in E ::= \quad & x \mid \mathsf{e} \mid \pi \mid 0 \mid 1 \mid 2 \mid \ldots \\
& \mid \log(e) \mid -e \mid e_1 + \ldots + e_n \mid e_1 \cdot \ldots \cdot e_n \mid e_1^{e_2} \\
& \mid \delta(e) \mid [e_1 = e_2] \mid [e_1 \leq e_2] \mid [e_1 \neq e_2] \mid [e_1 < e_2] \\
& \mid \int_{\mathbb{R}} \mathsf{d}x \, e[\![x]\!] \mid \sum_{x \in \mathbb{Z}} e[\![x]\!] \mid \varphi(e_1, \ldots, e_n) \\
& \mid (\mathsf{d}/\mathsf{d}x)^{-1}[\mathsf{e}^{-x^2}](e) \quad \text{(Error function)}
\end{aligned}
$$

Encodes probability density functions

$$\mathsf{Bernoulli}(x; p) = p \cdot \delta(1 - x) + (1 - p) \cdot \delta(x)$$

$$\mathsf{Gauss}(x; \mu, \nu) = [\nu = 0] \cdot \delta(x - \mu) + [\nu \neq 0] \cdot \frac{\mathsf{e}^{-(x-\mu)^2/(2\nu)}}{(2\pi\nu)^{\frac{1}{2}}}$$

$$\mathsf{UniformInt}(x; a, b) = \frac{\sum_{x' \in \mathbb{Z}} \delta(x - x') \cdot [a \leq x'] \cdot [x' \leq b]}{\sum_{x' \in \mathbb{Z}} [a \leq x'] \cdot [x' \leq b]}$$

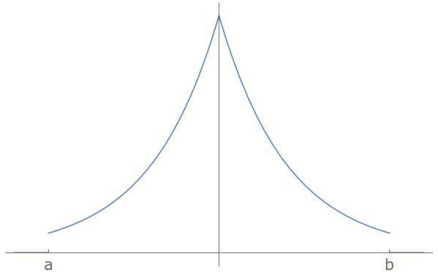
How About Sum of Three Variables?

$Z = X1 + X2 + X3$

X1, X2, X3
i.i.d.

Z

Truncated Laplace

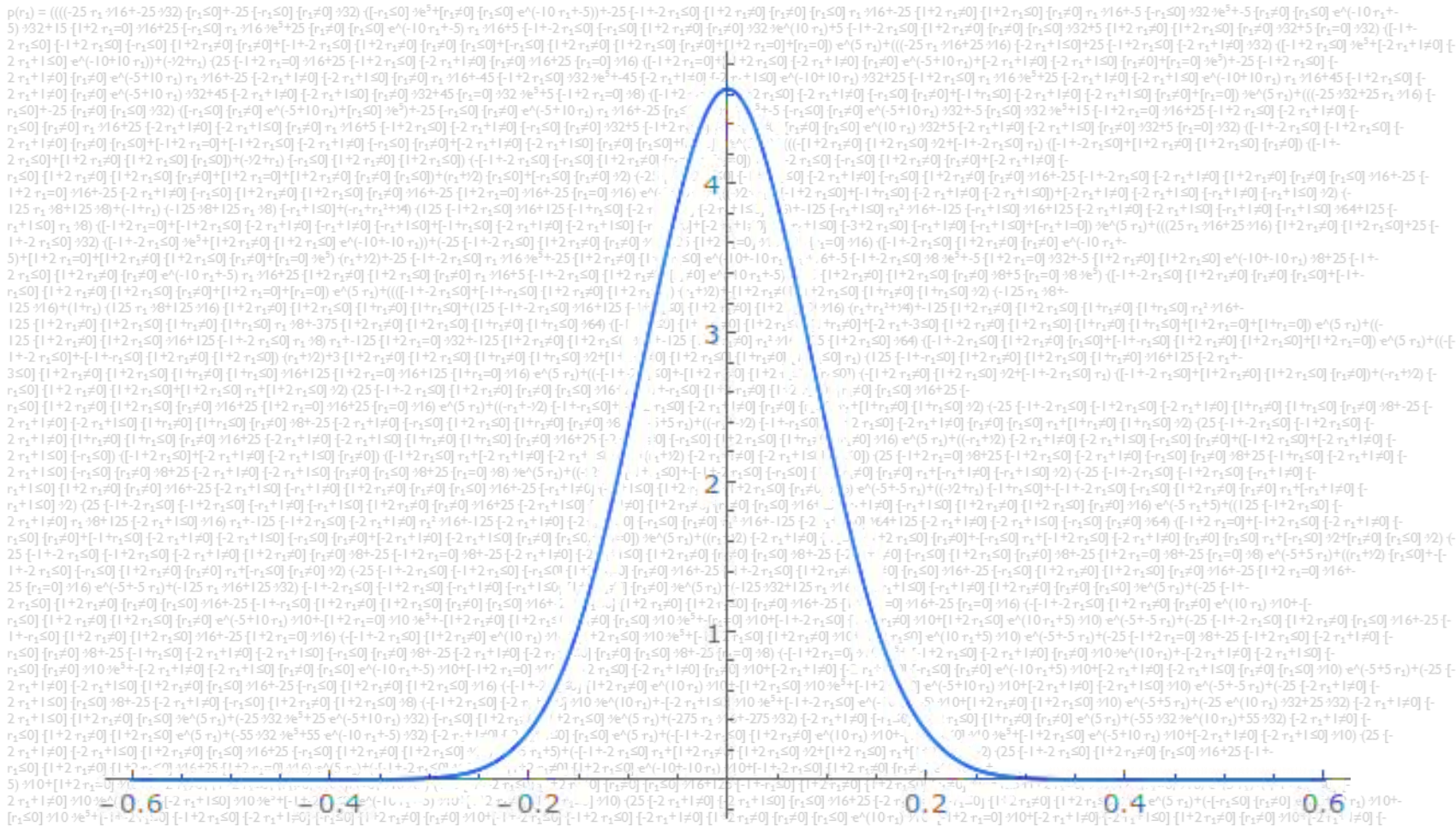


?

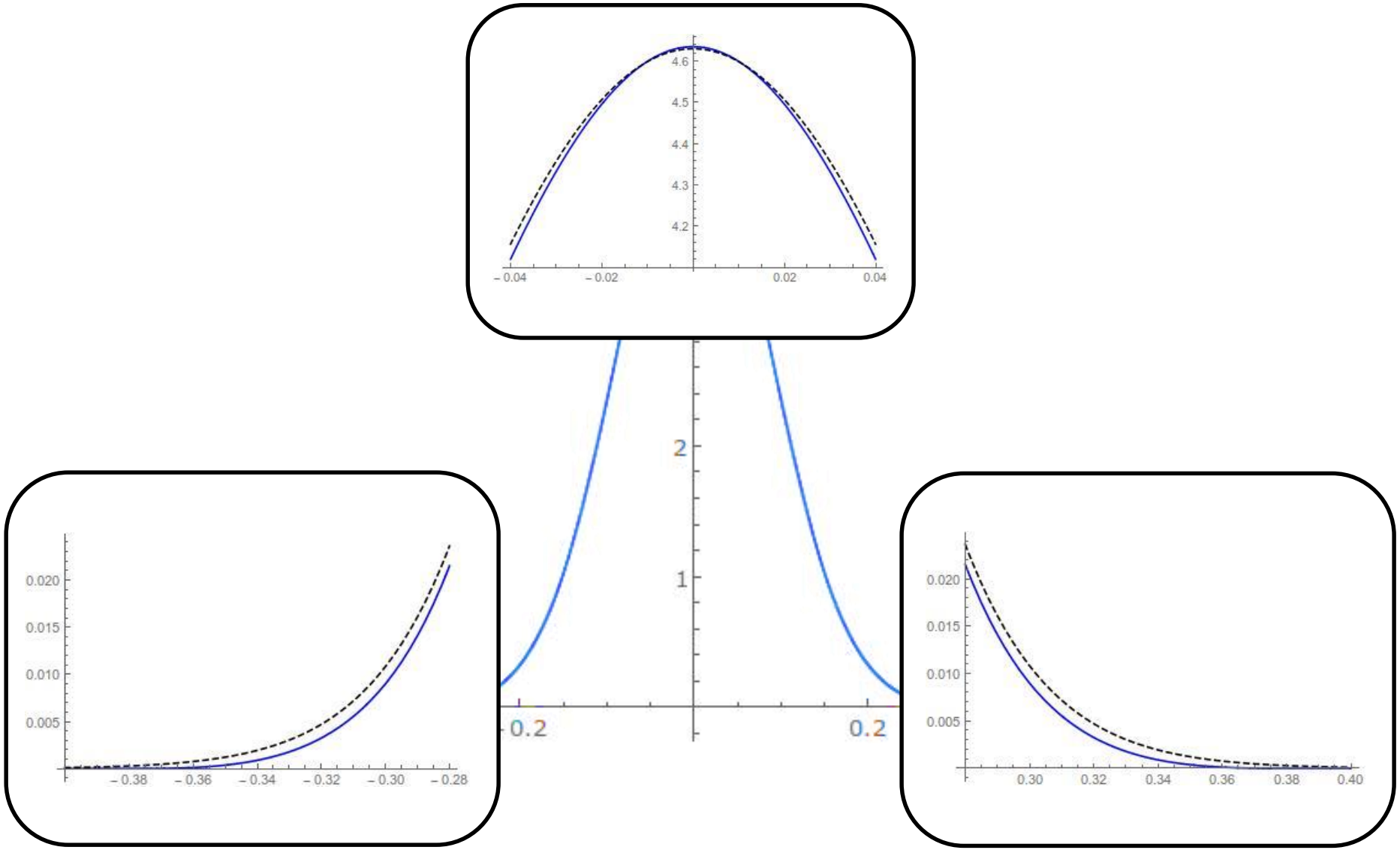
Exact Final Distribution (PSI)

p(r1) = ((((-25 r1 ⅟16+-25 ⅟32) [r1≤0]+-25 [-r1≤0] [r1≠0] ⅟32) ([-r1≤0] ⅇ^5+[r1≠0] [r1≤0] e^(-10 r1+-5))+-25 [-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] r1 ⅟16+-25 [1+2 r1≠0] [1+2 r1≤0] [r1≠0] r1 ⅟16+-5 [-r1≤0] ⅟32 ⅇ^5+-5 [r1≠0] [r1≤0] e^(-10 r1+-5) ⅟32+15 [1+2 r1=0] ⅟16+25 [-r1≤0] r1 ⅟16 ⅇ^5+25 [r1≠0] [r1≤0] e^(-10 r1+-5) r1 ⅟16+5 [-1+-2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0] ⅟32 ⅇ^(10 r1)+5 [-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟32+5 [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟32+5 [r1=0] ⅟32) ([-1+-2 r1≤0] [-1+2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0]+[-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0]+[-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0]+[1+2 r1=0]+[r1=0]) e^(5 r1)+(((-25 r1 ⅟16+25 ⅟16) [-2 r1+1≤0]+25 [-1+2 r1≤0] [-2 r1+1≠0] ⅟32) ([-1+2 r1≤0] ⅇ^5+[-2 r1+1≠0] [-2 r1+1≤0] e^(-10+10 r1))+(-⅟2+r1) (25 [-1+2 r1=0] ⅟16+25 [-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] ⅟16+25 [r1=0] ⅟16) ([-1+2 r1=0]+[-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] e^(-5+10 r1)+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0]+[r1=0] ⅇ^5)+-25 [-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] e^(-5+10 r1) r1 ⅟16+-25 [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] r1 ⅟16+-45 [-1+2 r1≤0] ⅟32 ⅇ^5+-45 [-2 r1+1≠0] [-2 r1+1≤0] e^(-10+10 r1) ⅟32+25 [-1+2 r1≤0] r1 ⅟16 ⅇ^5+25 [-2 r1+1≠0] [-2 r1+1≤0] e^(-10+10 r1) r1 ⅟16+45 [-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] e^(-5+10 r1) ⅟32+45 [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] ⅟32+45 [r1=0] ⅟32 ⅇ^5+5 [-1+2 r1=0] ⅟8) ([-1+2 r1=0]+[-1+2 r1≤0] [-2 r1+1≠0] [-r1≤0] [r1≠0]+[-1+r1≤0] [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0]+[r1=0]) ⅇ^(5 r1)+(((-25 ⅟32+25 r1 ⅟16) [-r1≤0]+-25 [r1≠0] [r1≤0] ⅟32) ([-r1≤0] [r1≠0] e^(-5+10 r1)+[r1≤0] ⅇ^5)+-25 [-r1≤0] [r1≠0] e^(-5+10 r1) r1 ⅟16+-25 [r1≤0] r1 ⅟16 ⅇ^5+-5 [-r1≤0] [r1≠0] e^(-5+10 r1) ⅟32+-5 [r1≤0] ⅟32 ⅇ^5+15 [-1+2 r1=0] ⅟16+25 [-1+2 r1≤0] [-2 r1+1≠0] [-r1≤0] [r1≠0] r1 ⅟16+25 [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] r1 ⅟16+5 [-1+2 r1≤0] [-2 r1+1≠0] [-r1≤0] [r1≠0] ⅟32+5 [-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] [r1≤0] e^(10 r1) ⅟32+5 [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] ⅟32+5 [r1=0] ⅟32) ([-1+-2 r1≤0] [-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] [r1≤0]+[-1+2 r1=0]+[-1+2 r1≤0] [-2 r1+1≠0] [-r1≤0] [r1≠0]+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0]+[r1=0]) ⅇ^(5 r1)+(((-[1+2 r1≠0] [1+2 r1≤0] ⅟2+[-1+-2 r1≤0] r1) ([-1+-2 r1≤0]+[1+2 r1≠0] [1+2 r1≤0] [r1≠0]) ([-1+-2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0]+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0]) ([-1+-2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0]+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0]) (-[-1+-2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0]+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0]) ⅟16+-25 [-1+-2 r1≤0] [-2 r1+1≠0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+-25 [-1+2 r1=0] ⅟16+-25 [-2 r1+1≠0] [-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟16+-25 [1+2 r1=0] ⅟16+-25 [r1=0] ⅟16) ⅇ^(-5+-5 r1)+(((-⅟2+r1) ([-1+2 r1≤0]+[-1+r1≤0] [-2 r1+1≠0] [-2 r1+1≤0])+[-2 r1+1≠0] [-2 r1+1≤0] [-r1+1≠0] [-r1+1≤0] ⅟2) (-125 r1 ⅟8+125 ⅟8) (-1+r1) (-125 ⅟8+125 r1 ⅟8) [-r1+1≤0]+(-r1+r1²+⅟4) (125 [-1+2 r1≤0] ⅟16+125 [-1+r1≤0] [-2 r1+1≠0] [-2 r1+1≤0] ⅟16)+-125 [-r1+1≤0] r1² ⅟16+-125 [-r1+1≤0] ⅟16+125 [-2 r1+1≠0] [-2 r1+1≤0] [-r1+1≠0] [-r1+1≤0] ⅟64+125 [-r1+1≤0] r1 ⅟8) ([-1+2 r1=0]+[-1+2 r1≤0] [-2 r1+1≠0] [-r1+1≠0] [-2 r1+1≤0] [-2 r1+1≠0] [-2 r1+1≤0] [-r1+1≠0]+[-r1+1=0]) ⅇ^(5 r1)+(((25 r1 ⅟16+25 ⅟16) [1+2 r1≠0] [1+2 r1≤0]+25 [-1+-2 r1≤0] ⅟32) ([-1+-2 r1≤0] ⅇ^5+[1+2 r1≠0] [1+2 r1≤0] e^(-10+-10 r1))+(-25 [-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] ⅟16+-25 [1+2 r1=0] ⅟16+-25 [r1=0] ⅟16) ([-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] e^(-10 r1+-5)+[1+2 r1=0]+[1+2 r1≠0] [1+2 r1≤0] [r1≠0]+[r1=0] ⅇ^5) (r1+⅟2)+-25 [-1+-2 r1≤0] r1 ⅟16 ⅇ^5+-25 [1+2 r1≠0] [1+2 r1≤0] e^(-10+-10 r1) r1 ⅟16+-5 [-1+-2 r1≤0] ⅟8 ⅇ^5+-5 [1+2 r1=0] ⅟32+-5 [1+2 r1≠0] [1+2 r1≤0] e^(-10+-10 r1) ⅟8+25 [-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] e^(-10 r1+-5) r1 ⅟16+25 [1+2 r1≠0] [1+2 r1≤0] [r1≠0] r1 ⅟16+5 [-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] e^(-10 r1+-5) ⅟8+5 [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟8+5 [r1=0] ⅟8 ⅇ^5) ([-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0]+[-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0]+[1+2 r1=0]+[r1=0]) e^(5 r1)+(((([-1+-2 r1≤0]+[-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0]) (r1+⅟2)+-[1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0] ⅟2) (-125 r1 ⅟8+-125 ⅟16)+(1+r1) (125 r1 ⅟8+125 ⅟16) [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0]+(125 [-1+-2 r1≤0] ⅟16+125 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] ⅟16) (r1+r1²+⅟4)+-125 [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0] r1² ⅟16+-125 [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0] ⅟64) ([-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0]+[-2 r1+-3≤0] [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0]+[1+2 r1=0]+[1+r1=0]) e^(5 r1)+(((-125 [1+2 r1≠0] [1+2 r1≤0] ⅟16+125 [-1+-2 r1≤0] r1 ⅟8) r1+-125 [1+2 r1=0] ⅟32+-125 [1+2 r1≠0] [1+2 r1≤0] ⅟64+-125 [1+2 r1≠0] r1² ⅟16+125 [1+2 r1≤0] ⅟64) ([-1+-2 r1≤0] [1+2 r1≠0] [r1≤0]+[-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0]+[1+2 r1=0]) e^(5 r1)+(((-[1+-2 r1≤0]+-[-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0]) (r1+⅟2)+3 [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0] ⅟2+[1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0] r1) (125 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] ⅟16+125 [-2 r1+3≤0] [1+2 r1≠0] [1+2 r1≤0] [1+r1≠0] [1+r1≤0] ⅟16+125 [1+2 r1=0] ⅟16+125 [1+r1=0] ⅟16) e^(5 r1)+(((-[-1+-2 r1≤0]+-[1+2 r1≠0] [1+2 r1≤0] [r1≤0]) (-[1+2 r1≠0] [1+2 r1≤0] ⅟2+[-1+-2 r1≤0] r1) ([-1+-2 r1≤0]+[1+2 r1≠0] [1+2 r1≤0] [r1≠0])+(-r1+⅟2) (-r1≤0] [1+2 r1≠0] [1+2 r1≤0]+[1+2 r1≤0] r1+[1+2 r1≤0] ⅟2) (25 [-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+25 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] [r1≤0] ⅟16+25 [-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟8+25 [-1+2 r1=0] ⅟16+25 [r1=0] ⅟16) ⅇ^(5 r1)+((-r1+-⅟2) [-1+-r1≤0]+[-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] [r1≤0] r1+[1+r1≠0] [1+r1≤0] ⅟2) (-25 [-1+-2 r1≤0] [-1+2 r1≤0] [-2 r1+1≠0] [1+r1≠0] [1+r1≤0] [r1≠0] ⅟8+-25 [-2 r1+1≠0] [-2 r1+1≤0] [1+r1≠0] [1+r1≤0] [r1≠0] ⅟8+-25 [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] ⅟16+25 [-2 r1+1≠0] [-2 r1+1≤0] [1+r1≠0] [1+r1≤0] [r1≠0] ⅟16+25 [-2 r1+1≠0] [-r1≤0] [1+2 r1≤0] [1+r1≠0] [r1≠0] ⅟16) ⅇ^(-5+5 r1)+((-r1+⅟2) [-1+r1≤0]+[-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] [r1≤0] r1+[1+r1≠0] [1+r1≤0] ⅟2) (25 [-1+-2 r1≤0] [-1+2 r1≤0] [-2 r1+1≠0] [1+r1≠0] [1+r1≤0] [r1≠0] ⅟8+25 [-2 r1+1≠0] [-2 r1+1≤0] [1+r1≠0] [1+r1≤0] [r1≠0] ⅟16+25 [-2 r1+1≠0] [-r1≤0] [1+2 r1≤0] [1+r1≠0] [r1≠0] ⅟16) ⅇ^(5 r1)+((-r1+⅟2) [-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0] [r1≠0]+([-1+2 r1≤0]+[-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0]) ([-1+2 r1≤0] r1+[-2 r1+1≠0] [-2 r1+1≤0] ⅟2)+(r1+⅟2) [-2 r1+1≠0] [-2 r1+1≤0] [r1≤0]) (25 [-1+2 r1=0] ⅟8+25 [-1+2 r1≤0] [-2 r1+1≠0] [-r1≤0] [r1≠0] ⅟8+25 [-1+r1≤0] [-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0] [r1≠0] ⅟8+25 [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] [r1≤0] ⅟8+25 [r1=0] ⅟8) ⅇ^(5 r1)+((-⅟2+r1) [-1+r1≤0]+[-1+2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0] r1+[-r1+1≠0] [-r1+1≤0] ⅟2) (-25 [-1+-2 r1≤0] [-1+2 r1≤0] [-r1+1≠0] [r1≠0] ⅟16+-25 [-2 r1+1≠0] [-r1+1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16) ⅇ^(-5+-5 r1)+((-⅟2+r1) [-1+r1≤0]+[-1+-2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0] r1+[-r1+1≠0] [-r1+1≤0] ⅟2) (25 [-1+-2 r1≤0] [-1+2 r1≤0] [-r1+1≠0] [-r1+1≤0] [1+2 r1≠0] [r1≠0] ⅟16+25 [-2 r1+1≠0] [-r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+25 [-r1+1≠0] [-r1+1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟16) ⅇ^(-5 r1+5)+((1 125 [-1+2 r1≤0] [-2 r1+1≠0] r1 ⅟8+125 [-2 r1+1≤0] ⅟16) r1+-125 [-1+2 r1≤0] [-2 r1+1≠0] r1² ⅟16+-125 [-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0] [r1≠0] r1² ⅟16+-125 [-2 r1+1≤0] ⅟64+125 [-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0] [r1≠0] ⅟64) ([-1+2 r1=0]+[-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] [-1+r1≤0] [-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0] [r1≠0]+[-r1≤0] r1+[-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] [r1≤0] ⅟2+[r1≠0] [r1≤0] ⅟2) (-25 [-1+-2 r1≤0] [-1+2 r1≤0] [-2 r1+1≠0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟8+-25 [-1+2 r1=0] ⅟8+-25 [-2 r1+1≠0] [-2 r1+1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟8+-25 [-2 r1+1≠0] [-2 r1+1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟8+-25 [1+2 r1=0] ⅟8+-25 [r1=0] ⅟8) ⅇ^(5 r1)+((r1+⅟2) [-2 r1+1≠0] [-r1≤0] [1+2 r1≠0] [r1≠0] r1+[-r1≤0] ⅟2+[r1≠0] [r1≤0] ⅟2) (-25 [-1+-2 r1≤0] [-1+2 r1≤0] [-2 r1+1≠0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟8+-25 [-2 r1+1≠0] [-2 r1+1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟8+-25 [-2 r1+1≠0] [-r1≤0] [1+2 r1≤0] [r1≠0] ⅟8+-25 [1+2 r1=0] ⅟8+-25 [r1=0] ⅟8) ⅇ^(-5+5 r1)+((r1+⅟2) [-r1≤0]+-[-1+-2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0] r1+[-r1≤0] [-r1≤0] ⅟2) (-25 [-1+-2 r1≤0] [-1+2 r1≤0] [-r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+-25 [-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+-25 [-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟16+-25 [1+2 r1=0] ⅟16+-25 [r1=0] ⅟16) ⅇ^(-5+-5 r1)+(-125 ⅟32+125 r1 ⅟16) [-2 r1+1≤0] [-r1+1≠0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟e^(5 r1)+(-25 [-1+-2 r1≤0] [-2 r1+1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+-25 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] [r1≤0] ⅟16+-25 [-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟16+-25 [1+2 r1=0] ⅟16+-25 [r1=0] ⅟16) ([-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] ⅟10+[-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+-25 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟16+-25 [1+2 r1=0] ⅟16+-25 [r1=0] ⅟16) ⅇ^(5 r1)+(-25 [-1+-2 r1≤0] [-2 r1+1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+-25 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] [r1≤0] ⅟16+-25 [-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟16+-25 [1+2 r1=0] ⅟16+-25 [r1=0] ⅟16) ([-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] ⅟10+[-1+-2 r1≤0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟10 ⅇ^5+-[r1=0] ⅟10+[-1+-2 r1≤0] [1+2 r1≠0] ⅟10+[1+2 r1≤0] e^(10 r1+5) ⅟10) e^(-5+-5 r1)+(-25 [-1+-2 r1≤0] [1+2 r1≠0] [r1≤0] ⅟16+-25 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] e^(-5+10 r1) ⅟10+-[1+2 r1=0] ⅟10 ⅇ^5+-[1+2 r1≠0] [1+2 r1≤0] [r1≠0] [r1≤0] ⅟10 ⅇ^5+-[r1=0] ⅟10+[-1+-2 r1≤0] [1+2 r1≠0] ⅟10+[1+2 r1≤0] e^(10 r1+5) ⅟10) e^(-5+-5 r1)+(-25 [-1+-2 r1≤0] [1+2 r1≠0] [r1≤0] ⅟8+-25 [-1+r1≤0] [-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0] [r1≠0] ⅟8+-25 [-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] [r1≤0] ⅟8+-25 [r1=0] ⅟8) ([-1+2 r1=0] ⅟10 ⅇ^5+[-1+2 r1≤0] [-2 r1+1≠0] [r1≠0] ⅟10 ⅇ^(10 r1)+[-2 r1+1≠0] [-2 r1+1≤0] [-r1≤0] [r1≠0] ⅟10 ⅇ^5+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] ⅟10 ⅇ^5+[-1+2 r1≤0] e^(-10 r1+-5) ⅟10+[1+2 r1≠0] [1+2 r1≤0] e^(10 r1) ⅟10+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] ⅟10 ⅇ^(-5+-5 r1)+(-25 [-1+2 r1≤0] [-2 r1+1≠0] [r1≤0] ⅟8+-25 [-2 r1+1≠0] [-2 r1+1≤0] [1+2 r1≠0] [r1≠0] ⅟10+[-1+2 r1≤0] [-2 r1+1≠0] [-r1≤0] [1+2 r1≠0] [r1≠0] ⅟10+[-2 r1+1≠0] [-2 r1+1≤0] [1+2 r1≠0] [r1≠0] ⅟10+[-2 r1+1≠0] [-2 r1+1≤0] [r1≠0] e^(10 r1+5) ⅟10+[-2 r1+1≠0] [1+2 r1≤0] [r1≠0] ⅟10 ⅇ^5+[-1+2 r1≤0] e^(-10 r1+-5) ⅟10+[1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟10 ⅇ^(10 r1)+[-1+2 r1≤0] (25 [-2 r1+1≠0] [-2 r1+1≤0] [r1≤0] ⅟16+25 [-2 r1+1≠0] [-2 r1+1≤0] [r1≤0] ⅟16) ⅇ^(5 r1)+(-[-r1≤0] ⅟10 ⅇ^5+-[r1≠0] [r1≤0] e^(-10 r1+-5) ⅟10+[-1+-2 r1≤0] [-2 r1+1≠0] [r1≠0] ⅟10 ⅇ^(10 r1)+[-1+-2 r1≤0] (25 [-1+-2 r1≤0] [-1+2 r1≤0] [-2 r1+1≠0] [1+2 r1≠0] [r1≠0] [r1≤0] ⅟16+25 [-1+-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] [r1≤0] ⅟16+25 [-r1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟16+25 [1+2 r1=0] ⅟16+25 [r1=0] ⅟16) ⅇ^(5 r1)+(275 r1 ⅟16+275 ⅟32) [-1+-2 r1≤0] [-1+2 r1≤0] [-2 r1+1≠0] [1+r1≠0] [1+r1≤0] [r1≠0] ⅇ^(5 r1)+125 [-r1+1≠0] [-r1+1≤0] [1+2 r1≠0] [1+2 r1≤0] [r1≠0] ⅟32 ⅇ^(5 r1)+275 [-2 r1+1≠0] [-2 r1+1≤0] [1+r1≠0] [1+r1≤0] [r1≠0] e^(5 r1) ⅟32) (-3 ⅇ^(5 ⅟2)+-ⅇ^(15 ⅟2)+1+3 ⅇ^5)

Exact Final Distribution (PSI)



Exact Final Distribution (PSI)

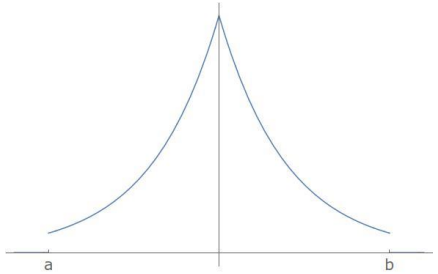


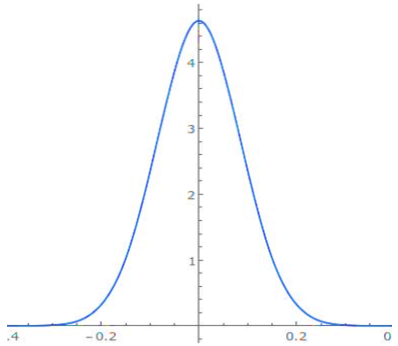
Different Input Assumptions

X1, X2, X3
i.i.d.

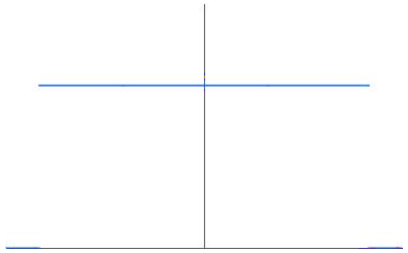
Z

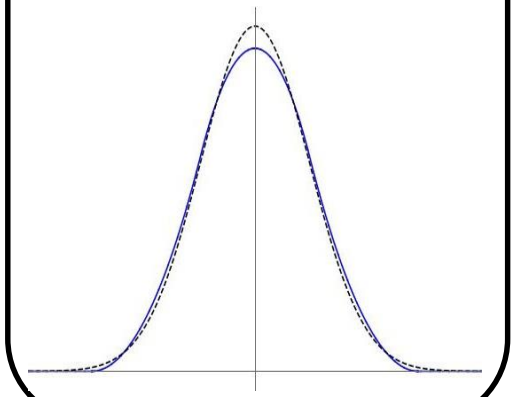
Truncated Laplace



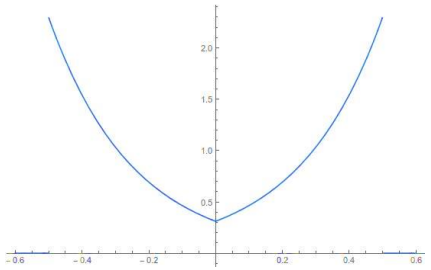


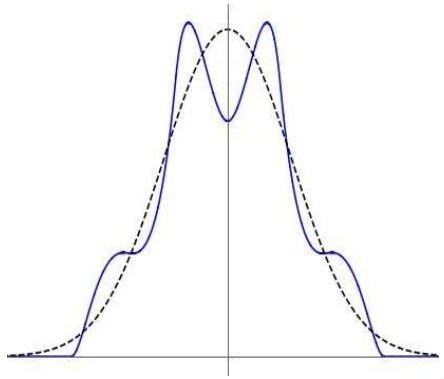
Uniform





Reverse Exponential





Continuous Example

```
X := Laplace(0,1)

Y := Laplace(0,1)

Z = X + Y

observe Z > 2

return X
```

Continuous Example

```
X := Laplace(0,1)

Y := Laplace(0,1)

Z = X + Y

observe Z > 2

return X
```

Variable Definition Rule:

0: Add variable to the state:

$$p() \rightarrow p(X)$$

1: Get the Laplace distribution expression

$$f(t_{tmp}) = \frac{\lambda}{2} e^{-\lambda|t_{tmp}-\mu|}$$

2: Relate it to the program's variable X

$$p(X) = \text{replace } t_{tmp} \text{ with } X \text{ in } f(t_{tmp})$$

Continuous Example

```
X := Laplace(0,1)

Y := Laplace(0,1)

Z = X + Y

observe Z > 2

return X
```

Variable Definition Rule:

0: Add variable to the state:

$$p() \rightarrow p(X)$$

1: Get the Laplace distribution expression

$$f(t_{tmp}) = \frac{\lambda}{2} e^{-\lambda|t_{tmp}-\mu|}$$

2: Relate it to the program's variable X

$$p(X) = \int dt_{tmp} \cdot \delta[X - t_{tmp}] \cdot f(t_{tmp})$$

Continuous Example

```
X := Laplace(0,1)
```

$$p(X) = \int dt_{tmp} \cdot \delta[X - t_{tmp}] \cdot \frac{\lambda}{2} e^{-\lambda|t_{tmp}-\mu|}$$

```
Y := Laplace(0,1)

Z = X + Y

observe Z > 2

return X
```

Integration with Dirac Deltas:

$$\int dt_{tmp} \cdot \delta[X - t_{tmp}] \cdot f(t_{tmp}) = f(X)$$

Continuous Example

```
X := Laplace(0,1)
```

$$p(X) = \frac{\lambda}{2} e^{-\lambda|X-\mu|}$$

```
Y := Laplace(0,1)

Z = X + Y

observe Z > 2

return X
```

Integration with Dirac Deltas:

$$\int dt_{tmp} \cdot \delta[X - t_{tmp}] \cdot f(t_{tmp}) = f(X)$$

Continuous Example

```
X := Laplace(0,1)
```

$$p(X) = \frac{1}{2} e^{-|X|}$$

```
Y := Laplace(0,1)

Z = X + Y

observe Z > 2

return X
```

Integration with Dirac Deltas:

$$\int dt_{tmp} \cdot \delta[X - t_{tmp}] \cdot f(t_{tmp}) = f(X)$$

Continuous Example

```
X := Laplace(0,1)
```

$$p(X) = \frac{1}{2}e^{-|X|}$$

```
Y := Laplace(0,1)
```

$$p(X,Y) = \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|}$$

```
Z = X + Y

observe Z > 2

return X
```

Continuous Example

```
X := Laplace(0,1)
```

$$p(X) = \frac{1}{2}e^{-|X|}$$

```
Y := Laplace(0,1)
```

$$p(X,Y) = \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|}$$

```
Z = X + Y
```

$$p(X,Y,Z) = \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|} \cdot \delta(Z - (X+Y))$$

```
observe Z > 2

return X
```

Assignment Rule (Deterministic):

0: Add variable Z to the state:

$$p(X,Y) \rightarrow p(X,Y,Z)$$

1: Relate the program's variables

$$p(X,Y,Z) = p(X,Y) \cdot \delta[Z - Expr(X,Y)]$$

Continuous Example

```
X := Laplace(0,1)
```

$$p(X) = \frac{1}{2}e^{-|X|}$$

```
Y := Laplace(0,1)
```

$$p(X,Y) = \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|}$$

```
Z = X + Y
```

$$p(X,Y,Z) = \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|} \cdot \delta(Z - (X+Y))$$

```
observe Z > 2
```

$$p(X,Y,Z|Z>2) \sim \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|} \cdot \delta(Z - (X+Y)) \cdot [Z > 2]$$

```
return X
```

Continuous Example

```
X := Laplace(0,1)
```

$$p(X) = \frac{1}{2}e^{-|X|}$$

```
Y := Laplace(0,1)
```

$$p(X,Y) = \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|}$$

```
Z = X + Y
```

$$p(X,Y,Z) = \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|} \cdot \delta(Z - (X+Y))$$

```
observe Z > 2
```

$$p(X,Y,Z|Z>2) \sim \frac{1}{2}e^{-|X|} \cdot \frac{1}{2}e^{-|Y|} \cdot \delta(Z - (X+Y)) \cdot [Z > 2]$$

```
return X
```

1. Marginalization
2. Normalization

$$p(X \mid Z > 2) \sim \int dY \int dZ \;\; p(X,Y,Z|Z>2)$$

$$Const = \int dX \int dY \int dZ \;\; p(X,Y,Z|Z>2)$$

Continuous Example

1. Marginalization

$$p(X \mid Z > 2) \sim \int dY \int dZ \;\; p(X, Y, Z | Z > 2)$$

$$\sim \int dY \int dZ \;\; \frac{1}{2} e^{-|X|} \cdot \frac{1}{2} e^{-|Y|} \cdot \delta(Z - (X + Y)) \cdot [Z > 2]$$

$$\sim \int dY \;\; \frac{1}{2} e^{-|X|} \cdot \frac{1}{2} e^{-|Y|} \cdot [X + Y > 2]$$

$$\sim \frac{1}{4} e^{-|X|} \cdot \left(e^{X-2} \cdot [X \leq 2] + \left(2 - e^{2-X}\right) \cdot [X > 2] \right)$$

2. Normalization

$$Const = \int dx \int dy \int dz \;\; p(X, Y, Z | Z > 2) = e^{-2}$$

Simplification Rules

Wolfram Mathematica 10, Maple™ ?

- Unpredictable
- Sometimes slow
- Sometimes wrong

PSI uses its own symbolic reasoning

Simplification Rules

- **Basic Algebraic Simplifications**

  $x + x \to 2 \cdot x, \quad x \cdot x \to x^2, \ldots$

- **Simplifications of Constraints**

  $[x = 0] + [x \neq 0] \to 1, \quad [x \leq 0] \cdot [0 \leq x] \to [x = 0],$

  $\delta(x) \cdot [1 \leq x] \to 0, \ldots$

- **Guard Linearization**

  $\delta(y - x^2) \to [-y \leq 0] \cdot ([x = 0] \cdot \delta(y) + [x \neq 0] \cdot \frac{1}{2\sqrt{y}}(\delta(x - \sqrt{y}) + \delta(x + \sqrt{y})))$

- **Symbolic Integration**

  $\int_{\mathbb{R}} \mathrm{d}x \int_{\mathbb{R}} \mathrm{d}y \, [0 \leq x] \cdot [x \leq 1] \cdot [0 \leq y] \cdot [y \leq 1] \cdot \delta(z - x \cdot y) \to$

  $-[0 < z] \cdot [z \leq 1] \cdot \log(z), \ldots$

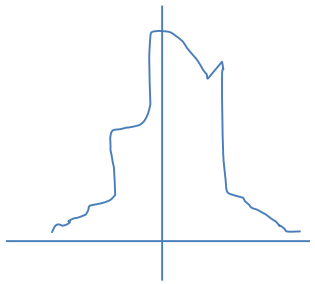
Approaches to Probabilistic Inference

Exact and Approximate

Sampling

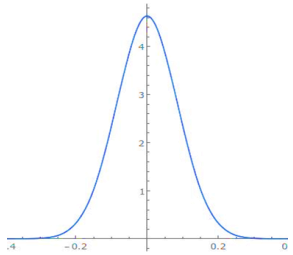
(Rejection – Church)

(MCMC – Church & Stan & R2)



Variational Inference

(Fun & Infer.NET)



Exact Symbolic Inference

(PSI & Hakaru)