

# Partial Operator Induction with Beta Distribution

---

Nil Geisweiller

AGI-18

SingularityNET

OpenCog Foundations

Problem:

Combining Models from Different Contexts

Theory:

Solomonoff Operator Induction and Beta Distribution

Practice:

Inference Control Meta-Learning

Problem:

Combining Models from Different Contexts

Theory:

Solomonoff Operator Induction and Beta Distribution

Practice:

Inference Control Meta-Learning

Problem:

Combining Models from Different Contexts

Theory:

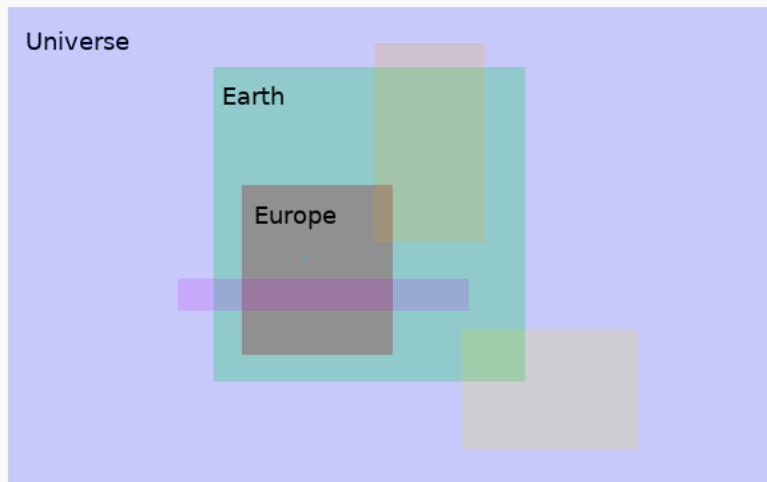
Solomonoff Operator Induction and Beta Distribution

Practice:

Inference Control Meta-Learning

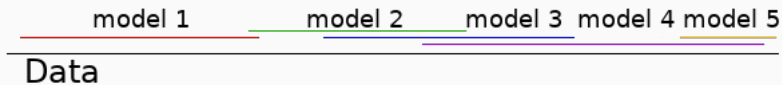
# Problem

How to combine models obtained from different contexts?

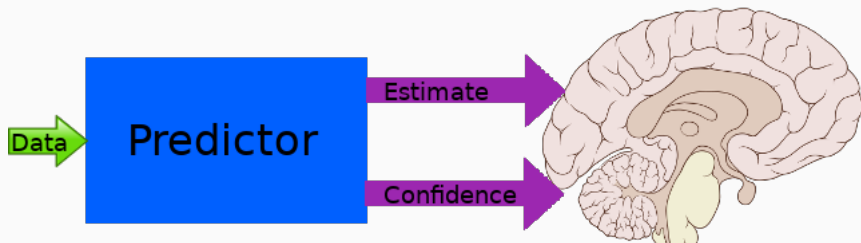


Bayesian Model Averaging (esp. Solomonoff Operator Induction)

+ partial models (obtained from different data sets)

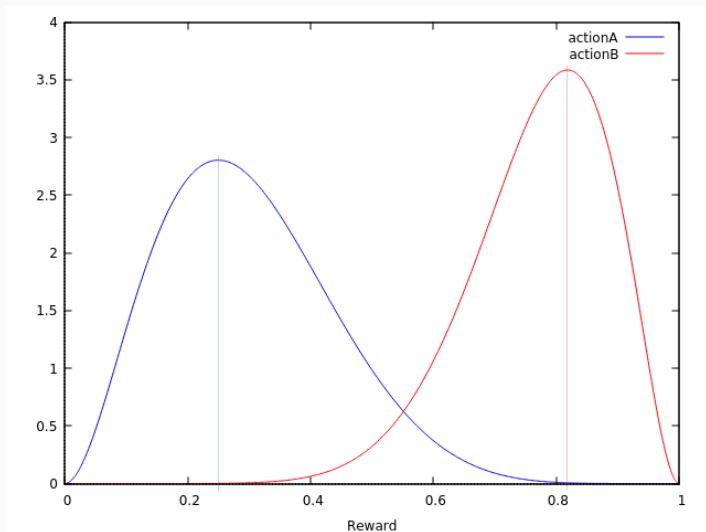


# Preserve Uncertainty



# Preserve Uncertainty

## Exploration vs Exploitation (Thompson Sampling)





Problem:

Combining Models from Different Contexts

Theory:

Solomonoff Operator Induction and Beta Distribution

Practice:

Inference Control Meta-Learning

Problem:

Combining Models from Different Contexts

Theory:

Solomonoff Operator Induction and Beta Distribution

Practice:

Inference Control Meta-Learning

# Solomonoff Operator Induction

Probability Estimate:

$$\hat{P}(A_{n+1}|Q_{n+1}) = \sum_j a_0^j \prod_{i=1}^{n+1} O^j(A_i|Q_i)$$

where:

- $Q_i = i^{th}$  question
- $A_i = i^{th}$  answer
- $O^j = j^{th}$  operator
- $a_0^j =$  prior of  $j^{th}$  operator

# Beta Distribution

Probability Density Function:

$$pdf_{\alpha,\beta}(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

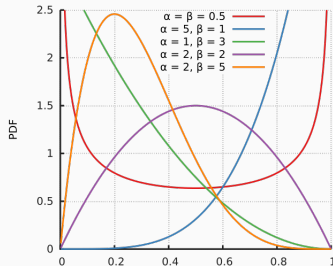
Beta Function:

$$B_x(\alpha, \beta) = \int_0^x p^{\alpha-1}(1-p)^{\beta-1} dp$$

$$B(\alpha, \beta) = B_1(\alpha, \beta)$$

Conjugate Prior:

$$pdf_{m+\alpha, n-m+\beta}(x) \propto x^m(1-x)^{n-m} pdf_{\alpha,\beta}(x)$$



# Beta Distribution Operator

OpenCog implication link

ImplicationLink <TV>

R

S

≡

Class of parameterized operators

$$O_p^j(A_i|Q_i) = \text{if } R^j(Q_i) \text{ then } \begin{cases} p, & \text{if } A_i = A_{n+1} \\ 1 - p, & \text{otherwise} \end{cases}$$

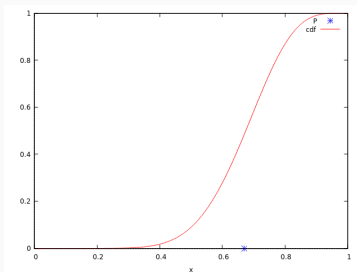
# Second Order Solomonoff Operator Induction

Probability Estimate:

$$\hat{P}(A_{n+1}|Q_{n+1}) = \sum_j a_0^j \prod_{i=1}^{n+1} O^j(A_i|Q_i)$$

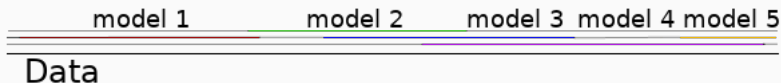
Probability Distribution Estimate:

$$\hat{cdf}_{(A_{n+1}|Q_{n+1})}(x) = \sum_{O^j(A_{n+1}|Q_{n+1}) \leq x} a_0^j \prod_{i=1}^n O^j(A_i|Q_i)$$



# Program Completion

$$O_{p,C}^j(A_i|Q_i) = \begin{cases} \text{if } R^j(Q_i) \text{ then } \begin{cases} p, & \text{if } A_i = A_{n+1} \\ 1 - p, & \text{otherwise} \end{cases} \\ \text{else } C(A_i|Q_i) \end{cases}$$



# Combing Solomonoff Operator Induction and Beta Distributions

$$\hat{cdf}_{(A_{n+1}|Q_{n+1})}(x) \propto \sum_j a_0^j r^j B_x(m^j + \alpha, n^j - m^j + \beta) B(m^j + \alpha, n^j - m^j + \beta)$$

where

- $n^j$  = number of observations explained by  $j^{th}$  model
- $m^j$  = number of true observations explained by  $j^{th}$  model
- $r^j$  = likelihood of the unexplained data

$$r^j = ???$$



# Combing Solomonoff Operator Induction and Beta Distributions

$$\hat{cdf}_{(A_{n+1}|Q_{n+1})}(x) \propto \sum_j a_0^j r^j B_x(m^j + \alpha, n^j - m^j + \beta) B(m^j + \alpha, n^j - m^j + \beta)$$

where

- $n^j$  = number of observations explained by  $j^{th}$  model
- $m^j$  = number of true observations explained by  $j^{th}$  model
- $r^j$  = likelihood of the unexplained data

$$r^j = ??? \approx 2^{-v^{(1-c)}}$$

- $v = n - n^j$  = number of unexplained observations
- $c$  = compressability parameter
  - $c = 1 \rightarrow$  explains remaining data
  - $c = 0 \rightarrow$  can't explain remaining data

Problem:

Combining Models from Different Contexts

Theory:

Solomonoff Operator Induction and Beta Distribution

Practice:

Inference Control Meta-Learning

Problem:

Combining Models from Different Contexts

Theory:

Solomonoff Operator Induction and Beta Distribution

Practice:

Inference Control Meta-Learning

# Inference Control Meta-learning

Learn how to reason efficiently

Methodology:

1. Solve sequence of problems (via reasoning)

# Inference Control Meta-learning

Learn how to reason efficiently

Methodology:

1. Solve sequence of problems (via reasoning)
2. Store inference traces

Learn how to reason efficiently

Methodology:

1. Solve sequence of problems (via reasoning)
2. Store inference traces
3. Mine traces to discover patterns

# Inference Control Meta-learning

Learn how to reason efficiently

Methodology:

1. Solve sequence of problems (via reasoning)
2. Store inference traces
3. Mine traces to discover patterns
4. Build control rules

Implication <TV>

And

<inference-pattern>

<rule>

<good-inference>

# Inference Control Meta-learning

Learn how to reason efficiently

Methodology:

1. Solve sequence of problems (via reasoning)
2. Store inference traces
3. Mine traces to discover patterns
4. Build control rules

Implication <TV>

And

<inference-pattern>

<rule>

<good-inference>

5. Combine control rules to guide future reasoning



# Combine Control Rules

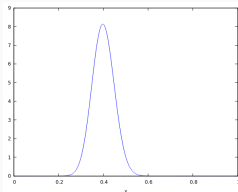
Implication <TV1>

And

<inference-pattern-1>

deduction-rule

<good-inference>



$c = 0.01$

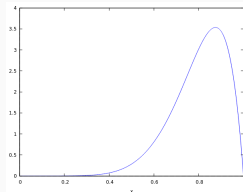
Implication <TV2>

And

<inference-pattern-2>

deduction-rule

<good-inference>



=

