AGI-20 Tutorial
OpenCog, PLN and Pattern Miner

Nil Geisweiller, Matt Ikle

SingularityNET & OpenCog Foundations

Table of Content:

1. Preparation

    10 *minutes*

2. Presentation

    20 *minutes*

3. Tutorial

    40 *minutes*

4. Discussion

1. Install docker
   - Debian/Ubuntu

     ```
     sudo apt install docker.io
     ```
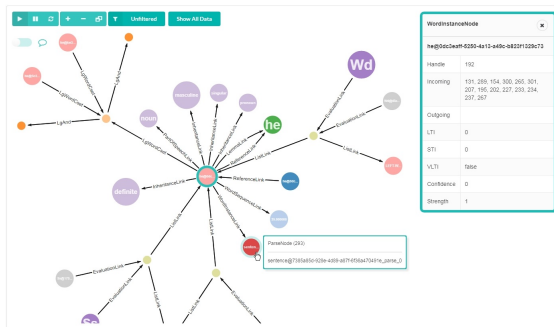   - Arch/Manjaro

     ```
     sudo pacman -S docker
     ```
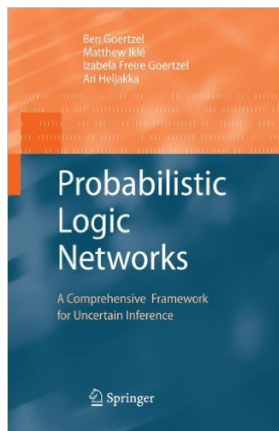2. Download docker image (1.6GB)

   ```
   sudo docker pull ngeiswei/opencog:agi20
   ```

Framework for AGI

1. Hypergraph Database:
   - AtomSpace
   - Atomese: query, rewrite and more
2. Mind Agents (cognitive processes):
   - Reasoning: PLN, Miner
   - Learning: MOSES, Miner
   - Decision: OpenPsi (Bach's MicroPsi)
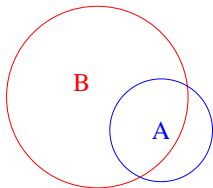   - Language Processing
   - Attention Allocation

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Probability Theory
- Uncertainty management
- Common sense reasoning
- Mathematical reasoning
- Resource management

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

```
(Subset (stv s c)
    A
    B)
```


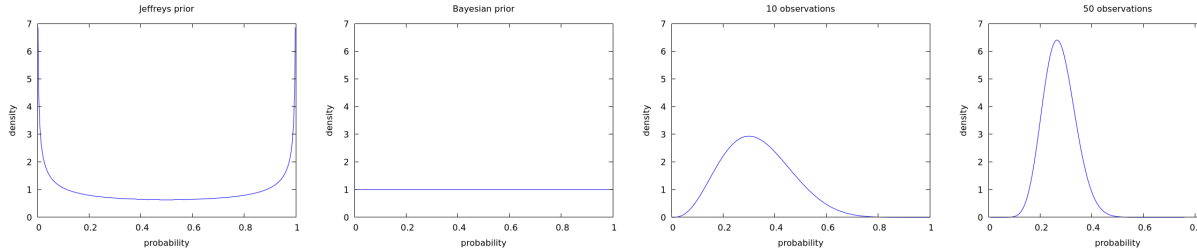**Definitions:**

- `stv` = *Simple Truth Value*
- `Subset A B` = $P(B|A)$
- `s` = strength = $P(B|A) = \dfrac{|A \cap B|}{|A|}$
- `c` = confidence = $\dfrac{|A|}{|A| + K}$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Truth Value = Second Order Distribution

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:
  ```
  (Subset (stv 0.4 0.1) A B)
  (Subset (stv 0.6 0.2) B C)
  A (stv 0.1 0.6)
  ```
- Rule base:

$$\frac{\text{(Subset <tv1> X Y)} \qquad \text{(Subset <tv2> Y Z)}}{\text{(Subset <tv3> X Z)}} \text{(Deduction)}$$

$$\frac{\text{(Subset <tv2> X Y)} \qquad \text{X <tv1>}}{\text{Y <tv3>}} \text{(Modus Ponens)}$$

- Target: `C <?>`

- -

  Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Unified Rule Engine

Rule base
Target

↓

Build inference tree
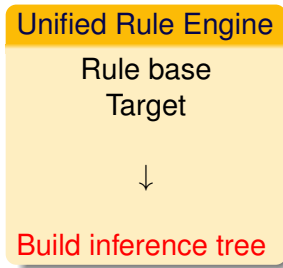
$$\frac{\dfrac{(\text{Subset} <?> A\ B) \qquad (\text{Subset} <?> B\ C)}{(\text{Subset} <?> A\ C)} (\text{Deduction}) \qquad A <?>}{C <?>} (\text{Modus Ponens})$$

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Basic:

- Deduction
- Modus Ponens
- Conjunction/Disjunction/Negation Introduction
- Universal Instantiation
- ...

Advanced:

- Intensional
- Contextual
- Temporal
- Spatial
- ...

Advanced built on basic

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Context-free control rule:

  ```
  (Implication <TV>
    <rule>
    <good-inference>)
  ```

- Context-sensitive control rule:

  ```
  (Implication <TV>
    (And
      <inference-context>
      <rule>)
    <good-inference>)
  ```

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:
  ```
  (Subset A B)
  (Subset A C)
  (Subset A D)
  (Subset B E)
  (Subset C E)
  (Subset D E)
  ...
  ```

- Frequent Pattern (minimum support = 3):

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:
  ```
  (Subset A B)
  (Subset A C)
  (Subset A D)
  (Subset B E)
  (Subset C E)
  (Subset D E)
  ...
  ```

- Frequent Pattern (minimum support = 3):
  ```
  (Subset A X)
  ```

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:
  ```
  (Subset A B)
  (Subset A C)
  (Subset A D)
  (Subset B E)
  (Subset C E)
  (Subset D E)
  ...
  ```

- Frequent Pattern (minimum support = 3):
  ```
  (Subset A X)
  (Subset X E)
  ```

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:
  ```
  (Subset A B)
  (Subset A C)
  (Subset A D)
  (Subset B E)
  (Subset C E)
  (Subset D E)
  ...
  ```

- Frequent Pattern (minimum support = 3):
  ```
  (Subset A X)
  (Subset X E)
  (And (Subset A X)  (Subset X E))
  ```

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Brute force algorithm:

- $\mathcal{K}$: *knowledge base*
- *S*: *minimum support*
- $\mathcal{C}$: *pattern pool*
- *P*, *Q*: *patterns*

1. Select *P* from $\mathcal{C}$
2. Select *specialization Q* of *P* such that $S \leq \mathrm{support}(Q, \mathcal{K})$
3. Add *Q* to $\mathcal{C}$
4. Repeat

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\frac{S \leq \mathrm{support}(Q, \mathcal{D}) \qquad \mathrm{spec}(Q, P)}{S \leq \mathrm{support}(P, \mathcal{D})} \text{ (AP=A Priory Property)}$$

$$\frac{S \leq \mathrm{support}(P, \mathcal{D}) \qquad \mathrm{spec}(P, \textit{Top})}{S \leq \mathrm{support}(\textit{Top}, \mathcal{D})} \text{ (AP)}$$
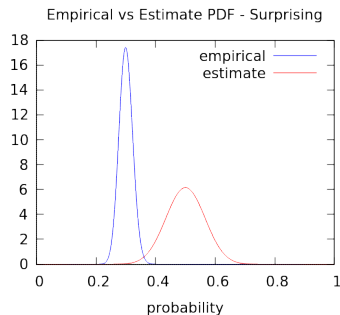
$$\Downarrow$$

$$\frac{\dfrac{S \leq \mathrm{support}(Q, \mathcal{D}) \qquad \mathrm{spec}(Q, P)}{S \leq \mathrm{support}(P, \mathcal{D})} \text{ (AP)} \qquad \mathrm{spec}(P, \textit{Top})}{S \leq \mathrm{support}(\textit{Top}, \mathcal{D})} \text{ (AP)}$$

$$\Downarrow$$

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

surprising: contrary to expectation

Empirical vs Estimate PDF - Surprising

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\frac{S \le \text{support}(P, \mathcal{D}) \quad \dfrac{\dfrac{P \quad \mathcal{D}}{\text{emp}(P, \mathcal{D})} \text{ (DE)} \quad \dfrac{?}{\text{est}(P, \mathcal{D})} \text{ (JSD)}}{\text{jsd}(P, \mathcal{D}))}}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)}$$

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\dfrac{S \leq \mathrm{support}(P,\mathcal{D}) \qquad \dfrac{\dfrac{P \qquad \mathcal{D}}{\mathrm{emp}(P,\mathcal{D})}\ \text{(DE)} \qquad \dfrac{P \qquad \mathcal{D}}{\mathrm{est}(P,\mathcal{D})}\ \text{(IS)}}{\mathrm{jsd}(P,\mathcal{D})}\ \text{(JSD)}}{\mathrm{surprising}(P,\mathcal{D},\mathrm{jsd}(P,\mathcal{D}))}\ \text{(S)}$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\dfrac{S \le \texttt{support}(P, \mathcal{D}) \qquad \dfrac{\dfrac{P \qquad \mathcal{D}}{\texttt{emp}(P, \mathcal{D})} \text{ (DE)} \qquad \dfrac{\vdots \qquad \vdots}{\texttt{est}(P, \mathcal{D})} \text{ (JSD)}}{\texttt{jsd}(P, \mathcal{D}))}}{\texttt{surprising}(P, \mathcal{D}, \texttt{jsd}(P, \mathcal{D}))} \text{ (S)}$$

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\cfrac{S \le \text{support}(P, \mathcal{D}) \qquad \cfrac{\cfrac{P \qquad \mathcal{D}}{\text{emp}(P, \mathcal{D})} \text{ (DE)} \qquad \cfrac{\begin{array}{c} \vdots \\ \cfrac{\text{emp}(Q, \mathcal{D})}{\vdots \qquad \vdots} \\ \vdots \qquad \vdots \end{array}}{\text{est}(P, \mathcal{D})} \text{ (JSD)}}{\text{jsd}(P, \mathcal{D}))}}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)}$$

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\cfrac{S \leq \texttt{support}(P,\mathcal{D}) \qquad \cfrac{\cfrac{P \qquad \mathcal{D}}{\texttt{emp}(P,\mathcal{D})}\text{(DE)} \qquad \cfrac{\cfrac{\cfrac{\texttt{emp}(Q,\mathcal{D})}{\vdots \qquad \vdots}}{\vdots \qquad \vdots}}{\texttt{est}(P,\mathcal{D})}\text{(JSD)}}{\texttt{jsd}(P,\mathcal{D}))}}{\texttt{surprising}(P,\mathcal{D},\texttt{jsd}(P,\mathcal{D}))}\text{(S)}$$

- -

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Tutorial time!

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Run docker image: `sudo docker run -it ngeiswei/opencog:agi20 bash`