

Python ACT-R Tutorials

- ▼ 1. Getting Started
 - 1. Install Python
 - 2. Install CCMSuite
 - 3. Run a model
- ▼ 2. Symbolic System
 - 1 – Simple production
 - 2 – Simple DM model
 - 3 – Simple instructions
- ▼ 3. Sub Symbolic System
 - 1 – Simple forgetting
 - 2 – Simple activation
 - 3 – Simple spreading
 - 4 – Simple partial
 - 5 – Simple inhibition
 - 6 – Simple utility learning
- ▼ 4. Environments
 - 1 – Simple environment
 - 2 – Simple agents
- ▼ 5. Vision
 - 1 – Simple vision
 - 2 – Top down system
- ▼ 6. Create Modules
 - 1 – Simple modules
- ▼ Reference Material
 - Associative Memory
 - Declarative Memory
 - Deconstructing ACT-R
 - Manual/Tutorial
 - Perceptual Motor
 - Python Productions
 - Utility algorithms

Sitemap

[Reference Material](#) >

Utility algorithms

Production Utility

Wed, 10/07/2009 - 12:30 — terry

Setting the utility of a production

By default, all productions start with a utility of 0. If you want to set this utility manually, rather than using a learning rule, you can do it like this:

```
def attendProbe(goal='state:start',vision='busy:False',location='?x ?y',utility=0.3):
```

Utility Learning Rules

The following learning rules exist for Python ACT-R

- ▣ PMPGC: the old PG-C learning rule

```
pm=PMPGC(goal=20)
```

- ▣ PMPGCSuccessWeighted: the success-weighted PG-C rule

```
pm=PMPGCSuccessWeighted(goal=20)
```

- ▣ PMPGCMixedWeighted: the mixed-weighted PG-C rule

```
pm=PMPGCMixedWeighted(goal=20)
```

- ▣ PMQLearn: standard Q-learning

```
pm=PMQLearn(alpha=0.2,gamma=0.9,initial=0)
```

- ▣ PMTD: TD-Learning (Fu & Anderson, 2004)

```
pm=PMQLearn(alpha=0.1,discount=1,cost=0.05)
```

- ▣ PMNew: the new ACT-R 6 standard learning rule

```
pm=PMNew(alpha=0.2)
```

Comments

You do not have permission to add comments.