Comparing Pavlik versus Weighted Trace model

Slim Stampen DataSet

Weighted Trace Model

Core idea:

$$A(t) = \log \sum_i w_i (t - t_i)^{-d}$$

Implementation:

- each trace has a weight w_i
- all traces share the same decay exponent d
- weights encode practice/surprisal
- activation feeds:

$$P = \sigma((A - \tau)/s)$$

$$RT = T_0 + Fe^{-A}$$

Pavlik model

Core idea:

$$A(t) = \log \sum_i (t - t_i)^{-d_i}$$

where:

$$d_i = \phi + ce^{A(t_i)}$$

Each trace has its own decay.

We fixed:

```ini
c = 0.5
```

Same decision rule:

$$P = \sigma((A - \tau)/s)$$

Same RT rule:

$$RT = T_0 + Fe^{-A}$$

How the data flow into the model

For each test trial:

1. **Replay all prior traces for that item**
2. **Compute activation A(t)**
3. **Predict accuracy**
4. **Predict RT**
5. **Compare to observed**
6. **Add log-likelihood**

From CSV headers:

```
subj
item
isCorrect
RT
time
trial
type
```

slimstampen_all_trials.csv

Open with

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	subj	item	resp	isCorrect	RT	duration	type	trial	time	levDist	rep	alpha	fixedRT	estRT
2	1501	3	fish	TRUE	2295	3.131	study	1	3.131	0	1	0.3	0.3	Inf
3	1501	3	fish	TRUE	888	2.295	test	2	5.426	0	2	0.3	0.3	1.7856
4	1501	12	book	TRUE	3128	4.822	study	3	10.248	0	1	0.3	0.3	Inf
5	1501	12	book	TRUE	841	2.528	test	4	12.776	0	2	0.3	0.3	1.9031
6	1501	2	beer	TRUE	4203	6.048	study	5	18.824	0	1	0.3	0.3	Inf
7	1501	9	police	TRUE	2456	4.989	study	6	23.813	0	1	0.3	0.3	Inf
8	1501	2	beer	TRUE	968	2.514	test	7	26.327	0	2	0.3	0.3	2.3551
9	1501	9	police	TRUE	1182	2.867	test	8	29.194	0	2	0.3	0.3	2.1304
10	1501	22	fruit	TRUE	2544	4.782	study	9	33.976	0	1	0.3	0.3	Inf
11	1501	5	office	TRUE	2128	4.596	study	10	38.572	0	1	0.3	0.3	Inf
12	1501	22	fruit	TRUE	1383	3.887	test	11	42.459	0	2	0.3	0.3	2.2572
13	1501	5	office	TRUE	1082	2.958	test	12	45.417	0	2	0.3	0.3	2.1991
14	1501	3	fish	TRUE	1634	3.055	test	13	48.472	0	3	0.2504	0.3	2.3399
15	1501	14	voice	TRUE	2391	3.926	study	14	52.398	0	1	0.3	0.3	Inf
16	1501	12	office	FALSE	2372	4.11	test	15	56.508	6	3	0.3496	0.3	2.3132
17	1501	14	voice	TRUE	3493	8.317	test	16	64.825	0	2	0.3	0.3	2.3772
18	1501	23	pain	TRUE	2084	4.342	study	17	69.167	0	1	0.3	0.3	Inf
19	1501	8	music	TRUE	1842	3.739	study	18	72.906	0	1	0.3	0.3	Inf
20	1501	23	pain	TRUE	1735	3.595	test	19	76.501	0	2	0.3	0.3	2.1716
21	1501	8	voice	FALSE	7976	9.31	test	20	85.811	4	2	0.3	0.3	2.118
22	1501	9	book	FALSE	2497	7.709	test	21	93.52	5	3	0.3496	0.3	2.557
23	1501	2	fruit	FALSE	1489	6.791	test	22	100.311	5	3	0.3496	0.3	2.6351
24	1501	22	fruit	TRUE	1693	6.469	test	23	106.78	0	3	0.2504	0.3	2.5629
25	1501	5	book	FALSE	2660	4.215	test	24	110.995	6	3	0.3496	0.3	2.5623
26	1501	12	book	TRUE	5512	10.199	test	25	121.194	0	4	0.3992	0.3	2.6203
27	1501	14	voice	TRUE	1570	2.915	test	26	124.109	0	3	0.2504	0.3	2.5043
28	1501	23	pain	TRUE	1298	2.478	test	27	126.587	0	3	0.2504	0.3	2.4037
29	1501	8	music	TRUE	4785	6.343	test	28	132.93	0	3	0.3496	0.3	2.3959

Model (the theory / equations)

These parts define what the model is and how it generates predictions.

- **Block 2) Helpers**
  `logistic()` and `softplus()` — math utilities used by the model.

- **Block 3) Replay: weighted traces → activation → p(correct) + RT**
  `replay_weighted_actr(...)` — this is the model simulator.
  It encodes:

  - activation equation (log sum of weighted traces)
  - surprisal weight rule
  - accuracy link (logistic)
  - RT link (T0 + F·exp(-A))

Subj

Used to split per participant.

`df[df["subj"] == subject_id]`

item

Memory identity.

Each item has its own trace history.

Model assumes:

separate memory system per item

Traces are stored per item.

Rules:

```
study → add trace
test + correct → add trace
test → generate prediction
```

This matches ACT-R learning assumptions.

time

This is the backbone of the model.

We use:

`dt = current_time - trace_time`

in the forgetting equation. Spacing effect is entirely driven by this column. If `time` is wrong → model collapses.

RT

Used only on test trials.

Converted:

`RT_sec = RT / 1000`

Then compared to:

`RT_pred = T0 + F * exp(-A)`

We penalize:

`log(RT_obs) - log(RT_pred)`

type

Two types:

```
study
test
```

We use this to decide:

- when to add traces
- when to evaluate predictions

isCorrect

This feeds the likelihood.

For test trials:

`p_correct = logistic(...)`

Then:

```
log likelihood = Bernoulli(p_correct
vs observed correctness)
```

This is the accuracy component of the NLL.

Fitted parameters example

```
[d, w1, tau, s, T0, F, sigma_rt]
[0.4568, 1.2898, -1.9468, 0.5911, 1.3987, 0.1352, 0.5661]
```

Memory parameters

d = 0.456

→ forgetting exponent
→ classic ACT-R range is ~0.3–0.7

w1 = 1.29

→ weight of the first trace
→ how surprising a new item is

Higher = stronger initial learning.

RT parameters

T0 = 1.40 sec

F = 0.135

How much activation affects RT; Scaling factor for memory speed.

Smaller F = memory strength influences RT less.

sigma_rt = 0.566

RT noise.

Human RT is noisy → this is normal.

Retrieval decision parameters

tau = -1.95

→ retrieval threshold

Negative threshold = easy retrieval.

Because it’s negative, the system is permissive: retrieval happens easily → high accuracy overall

s = 0.59

How sharp the accuracy curve is.

→ noise scale

Smaller = sharper decision boundary
This means performance changes steeply with activation.

Result Summary

Across subjects:

- Pavlik NLL ≈ 60–80 range
- Weighted NLL ≈ 150–200 range

That suggests:

forgetting-rate adaptation > weighted traces for this dataset.