Math Modeling, Week 1

A simple learning model

RL, prediction error, error correction

$\delta = R - P$

$P' = P + \varepsilon \cdot \delta$ or $\Delta P = \varepsilon \cdot \delta$

P is expectation (prediction), R is outcome (reward), δ is prediction error, ε is learning rate (internal parameter)

Examples

continuous outcomes: time (travel), reward (amount of food), punishment (pain, temperature)

discrete (binary) outcomes: event or no (food, shock), category A/B

→ prediction as probability

Mathematical expression of a verbal theory

What can we do with it?

- Formal derivation: predictions
- Elaborate it: incorporate other theoretical principles
  - Models aren't atomic!
- Simulate
- Evaluate fit to data
- Estimate parameters
- Formulate and test variants embodying competing hypotheses
- Use as measurement device
- Test experimental effects on parameter values

Formal predictions

Constant outcome (R)

$\Delta P = \varepsilon \cdot (R - P)$

Equilibrium: no change if $P = R$

Rate of approach: $Z = P - R$ (deviation). $\Delta Z = \Delta P - \Delta R = -\varepsilon Z$. $Z' = (1-\varepsilon)Z$

→ converges to correct value (R) exponentially, with rate parameter $1-\varepsilon$

Binary outcome, IID Bernoulli

Outcome as $\{0,1\}$

Rewarded (1) trials: $\Delta P = \varepsilon(1-P)$

$Z = P-1$, $Z' = (1-\varepsilon)Z$ → convergence to $Z=0$, P=1

Non-rewarded (0) trials: $\Delta P = -\varepsilon P$

$P' = (1-\varepsilon)P$, convergence to $P=0$

Mixture, $\Pr[R=1] = \alpha$

$\langle \Delta P \rangle = \alpha \cdot \varepsilon(1-P) + (1-\alpha) \cdot \varepsilon(0-P) = \varepsilon[\alpha \cdot 1 + (1-\alpha) \cdot 0 - P] = \varepsilon[\alpha - P]$

$\langle \Delta P \rangle = \varepsilon \langle \delta \rangle = \varepsilon[\langle R \rangle - P] = \varepsilon[\alpha - P]$

equilibrium, $\langle \Delta P \rangle = 0$, at $P = \langle R \rangle = \alpha$

same exponential convergence, *in the mean*, but also local sequential effects

tangent on annealing

Elaborate

RL is model of learning process

Add variable stimuli, and a model of representation

Feature decomposition, with additive association weights

$\mathbf{S} = [S_1, \ldots, S_n]$

$P = \mathbf{S} \cdot \mathbf{w} = \sum_i S_i \cdot w_i$

$\Delta w_i = \varepsilon \delta S_i$ (gradient descent: update each w_i in proportion to its contribution)

Rescorla-Wagner (1972): RL ∪ additive feature associations ∪ gradient descent

Simulation

Core matlab code

```
for t=1:n                                   %loop through trials
    p(t) = s(t,:)*w(:,t);                   %expected outcome
    delta = r(t) - p(t);                    %prediction error
    w(:,t+1) = w(:,t) + e*delta*s(t,:)';    %learning update
end
```

2 cues, binary outcome

Probability matching for response rule: $\Pr[r = 1] = P$

Plot of weight dynamics and response probability for a few cue designs:

- Blocking
- Two partially predictive cues
- One relevant and one irrelevant cue

Fit to data

Likelihood of data, according to model

Gives a number to quantify model fit (other methods too, e.g. SSE)

$\Pr[\mathbf{R} \mid \text{model}] = \prod_i \Pr[R_i \mid \text{model}]$

$\ln \Pr[\mathbf{R} \mid \text{model}] = \sum_i \ln \Pr[R_i \mid \text{model}]$

Compare model predictions to hypothetical data (graph).

How good? Hard to interpret in vacuum.

Estimate parameters

Plot learning rate vs loglikelihood

Peak is best-fitting model

Can get CI or standard error too

$\chi^2(1)$ distribution for difference in 2·loglikelihood

Test variants

Separate learning rates

$\Delta w_i = \varepsilon_i \delta S_i$

More free parameters (one per cue)

Necessarily fits better; significantly so?

Measurement

Psychological interpretation of parameters

Parameter estimate treated as a measurement—data transformation

Analogy: d'

Effects on parameter values

Comparing conditions or populations

Alternative to comparisons of raw behavior (%correct etc)

Compare estimated ε between groups

Often more valid

Less noise, process-pure

Standard statistics (t-test etc) on parameter estimates (2-step analysis)

Or hierarchical analysis: Takes likelihoods of model into account (accurate error theory)

Exercises

1. Play with the code

a) Execute the 4 blocks of code in order and look at the graphical results.

b) Change the paradigm (1, 2, or 3), and the learning rate, and explore how things change.

c) Test the model on a different paradigm (i.e., cue-outcome schedule), or try something else creative.

2. A pathology with high learning rates

a) What happens if $\varepsilon > ½$? Why? (Hint: simulate the model and then look at the values of p.)

b) That was for 2 cues. In general, with k cues, how large can ε be before the same pathology appears? How could the model be modified to avoid this problem?

3. Separate learning rates

a) Modify the code to allow a separate learning rate for each cue.

b) Generate data by simulating the common-ε model, then fit it using both the common-ε and the separate-ε models. The latter will involve a joint search over ε_i for all i (I suggest limiting to 2 cues). How much better does the separate-ε model fit?

c) Write a loop around steps 3a and 3b, to generate a sampling distribution of the difference in loglikelihood between the two models. What can you observe about this distribution?