NEU501: Learning & Memory

Lecture 3: action learning
trial by trial model fitting
model comparison

Act 1: where were we?
(action selection)

Temporal Difference (TD) learning

The problem: optimal prediction of future reinforcement

The algorithm:

$$V_t = E\left[r_{t+1}\right] + V_{t+1}$$

$$V_t^{T+1} = V_t^T + \eta \underbrace{\left(r_{t+1}^T + V_{t+1}^T - V_t^T\right)}_{\text{temporal difference prediction error } \delta(t+1)}$$

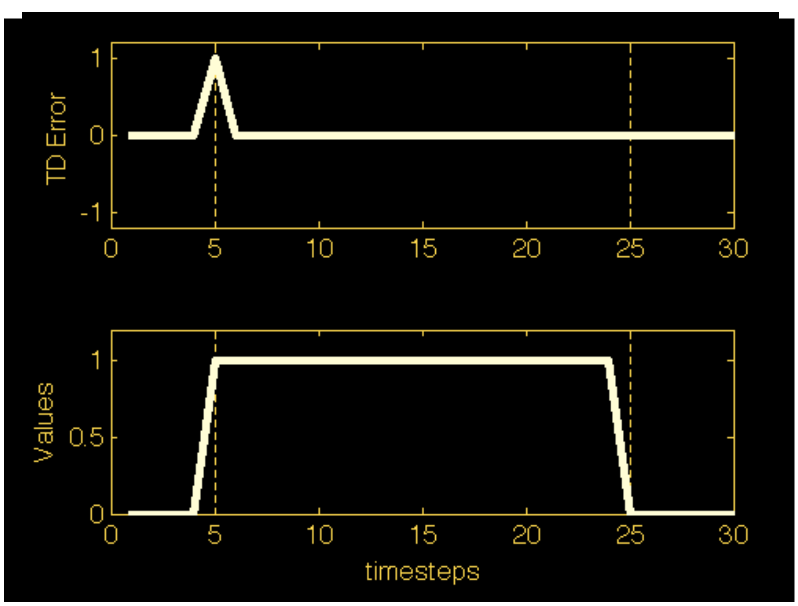
(note: *t* indexes time within a trial, T indexes trials)

compare to:

$$V^{T+1} = V^T + \eta\left(r^T - V^T\right)$$

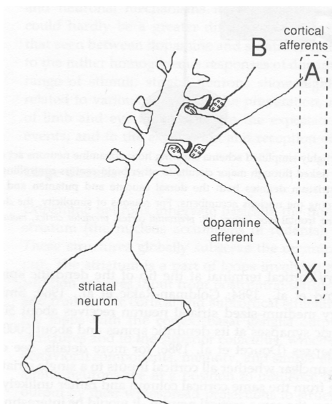
Sutton & Barto 1983, 1990

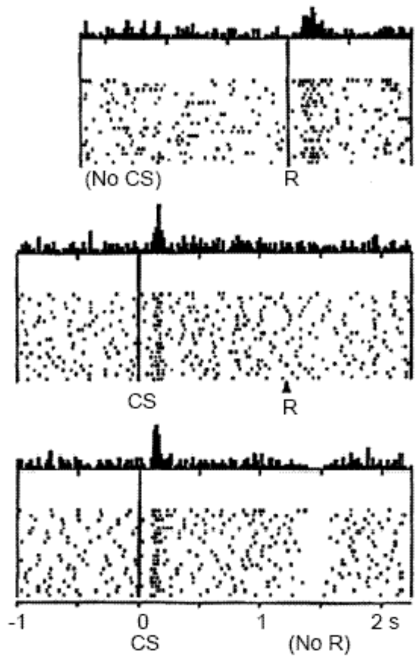
simulation

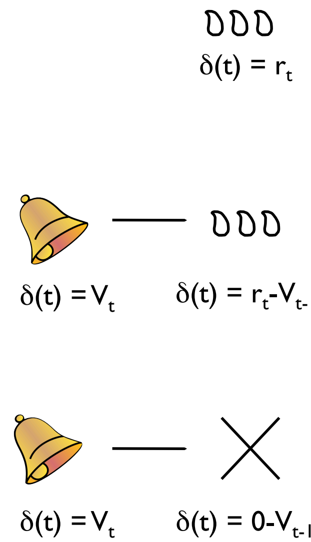


what would happen with partial reinforcement?
what would happen in second order conditioning?

implemented through dopamine



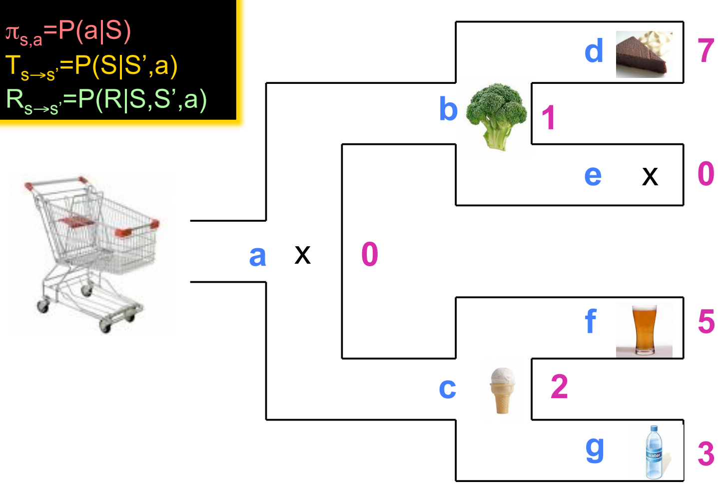




Schultz et al, 1997

MDPs

- States S
- Actions $\pi_{s,a}$=P(a|S)
- Transitions $T_{s \to s'}$=P(S|S',a)
- Rewards $R_{s \to s'}$=P(R|S,S',a)



- The idea: given the current situation, history does not matter
- $P(S_{t+1}|S_t,a_t) = P(S_{t+1}|S_1,S_2,\ldots,S_t,a_1,a_2,\ldots,a_t)$
- $P(r_t|S_t,a_t) = P(r_t|S_1,S_2,\ldots,S_t,a_1,a_2,\ldots,a_t)$

Stylized task: described fully by S,A,R,T

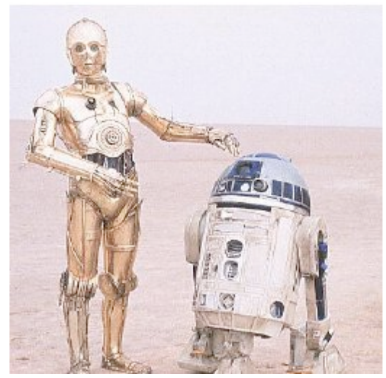
World: "You are in state 34. Your immediate reward is 3. You have 2 actions"

Robot: "I'll take action 1"

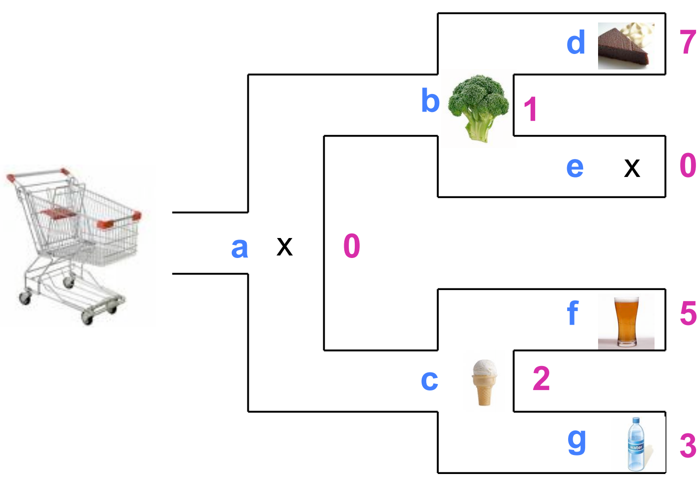
World: "You are in state 77. Your immediate reward is -7. You have 3 actions"

Robot: "I'll take action 3"

The task description requires no memory (*doesn't* mean that the decision maker does not use memory to solve the task!)



what can we compute here?



state values: $V(S)$ = E[sum of future rewards|S]

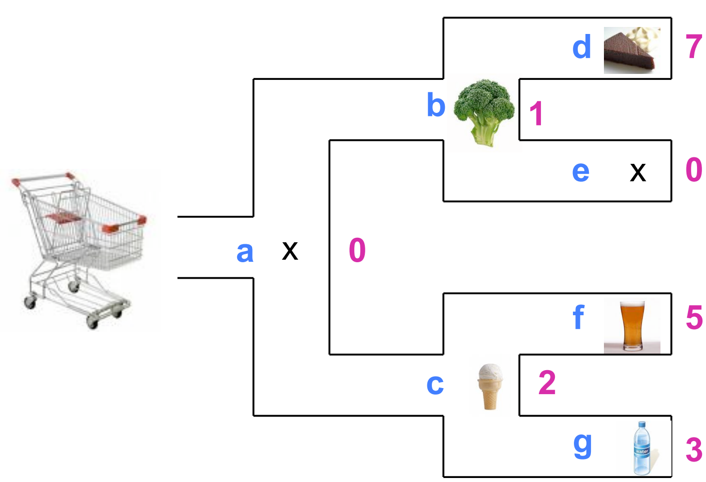
actually: $V^{\pi}(S)$ = E[sum of future rewards|π,S]

Key RL idea #1: Bellman's glorious equation

$$V^{\pi}(S) = \sum_a \pi_{s,a} \sum_{s'} T^a_{s \to s'} [R^a_{s \to s'} + V^{\pi}(S')]$$

In a Markov decision process state values are recursive

but there's more: computing the value of actions



(policy dependent) State-Action values:

Q^{π}(action|state) = E[sum of future rewards|S,a,π]

- Q(left|a) = ? Q(right|a) = ?
- which action is better?

Key RL idea #1 (again): Bellman's glorious equation

$$Q(S,a) = \sum_{s'} T^{a}_{s\rightarrow s'}[R^{a}_{s\rightarrow s'}+V(S')]$$

But.. what if we don't know T, R?

model-free learning: sampling

World: "You are in state 34. Your immediate reward is 3. You have 2 actions"

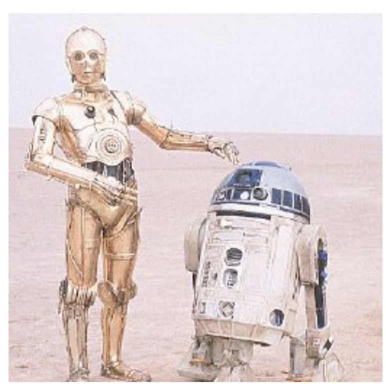
Robot: "I'll take action 1"

World: "You are in state 77. Your immediate reward is -7. You have 3 actions"

Robot: "I'll take action 3"

Take actions according to policy.

Treat experienced rewards and transitions as *samples*



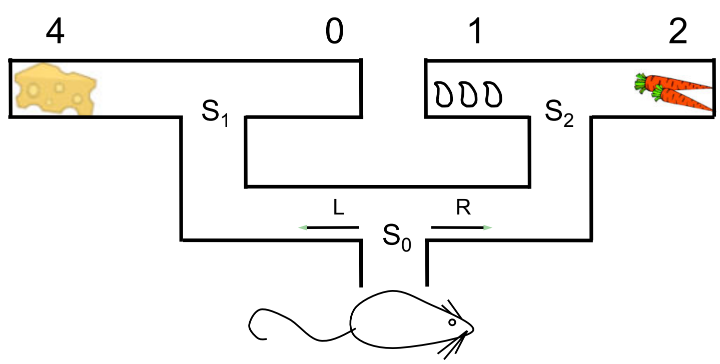
Key RL idea #2: Model free learning

$$V^{\pi}(S) = \sum_a \pi_{s,a} \sum_{s'} T^a_{s \to s'} [R^a_{s \to s'} + V^{\pi}(S')]$$

1. choose initial values $V_0(S)$
2. at time point t and state S_t behave according to π
3. observe S_{t+1} and $r(S_{t+1})$
4. compute prediction error $r(S_{t+1}) + V(S_{t+1}) - V(S_t)$
5. update $V(S_t)$ according to prediction error

learning of long-term values can be done using local information only

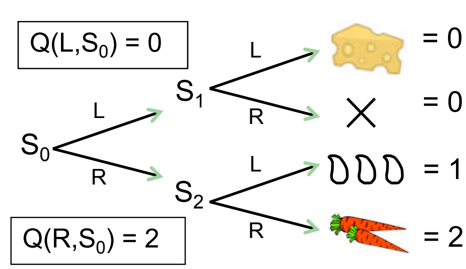
strategy 1: "model-based" RL



learn model of task through experience (= cognitive map)

compute Q values by "looking ahead" in the map

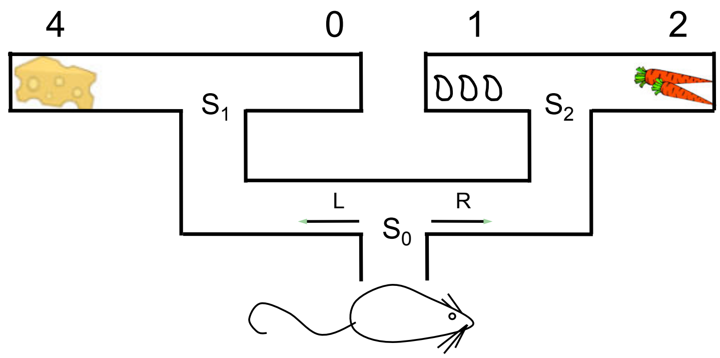
computationally costly, but also flexible (immediately sensitive to change)



Daw et al (2005)

14

strategy II: "model-free" RL



- Shortcut: store long-term values
  - then simply retrieve them to choose action
- Can learn these from experience
  - without building or searching a model
  - incrementally through prediction errors
  - dopamine dependent SARSA/Q-learning or Actor/Critic

Stored:

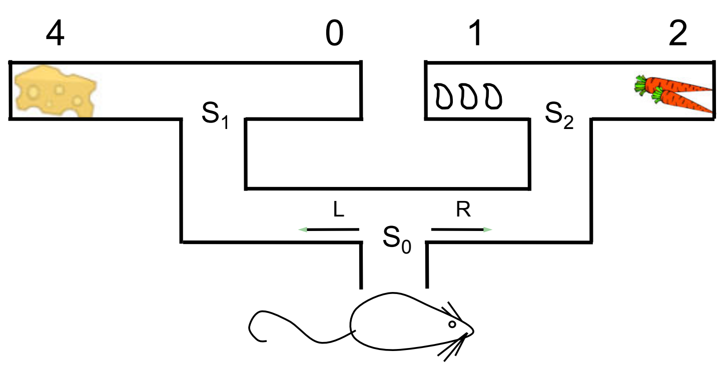
$Q(S_0,L) = 4$
$Q(S_0,R) = 2$

$Q(S_1,L) = 4$
$Q(S_1,R) = 0$

$Q(S_2,L) = 1$
$Q(S_2,R) = 2$

Daw et al (2005)

strategy II: "model-free" RL



- choosing actions is easy so behavior is quick, reflexive (S-R)
- but needs a lot of experience to learn
- and inflexible, need relearning to adapt to any change (habitual)

Stored:

$Q(S_0,L) = 4$
$Q(S_0,R) = 2$

$Q(S_1,L) = 4$
$Q(S_1,R) = 0$

$Q(S_2,L) = 1$
$Q(S_2,R) = 2$

Daw et al (2005)

summary so far

Instrumental learning: an instance of learning optimal control

MDPs: class of stylized tasks

In a Markov process long term values can be defined that

- are self consistent (recursively defined)
- can be learned incrementally (dynamic programming)
- can be learned from experience even without a world model

These values are helpful because they can help us improve the policy!

Act II: What does my *model* tell me about my *data*?

bandit tasks

overall approach:

- learn values for options (how is this problem simpler than those we've been talking about?)
- choose the best option

suppose we ran this experiment on a person:

- what are the data?
- what do our models predict?
- what can we conclude/infer from the data?

our models are basically detailed hypotheses about behavior and about the brain… we can test these hypotheses!

Writing down a full model of learning

what do we know?
what can we measure?
what do we not know?

Estimating model parameters

why estimate parameters?

1. May measure quantities of interest (learning rates in different populations, how variance in the task affects learning rate etc.)
2. have to use these to generate hidden variables of interest (eg. prediction errors) in order to look for these in the brain

how to estimate parameters?

we want: $P(\alpha,\beta \mid D,M)$

wwBd?

$P(\alpha,\beta \mid D,M) \propto P(D \mid \alpha,\beta,M)$ $\Longleftarrow$ This we know!

$P(D \mid \alpha,\beta,M) = \prod P(c_t \mid \alpha,\beta,M)$

Estimating model parameters

$P(\alpha,\beta \mid D,M) \propto P(D \mid \alpha,\beta,M)$

...this is a probability distribution

we often consider a point estimate: the maximal likelihood point

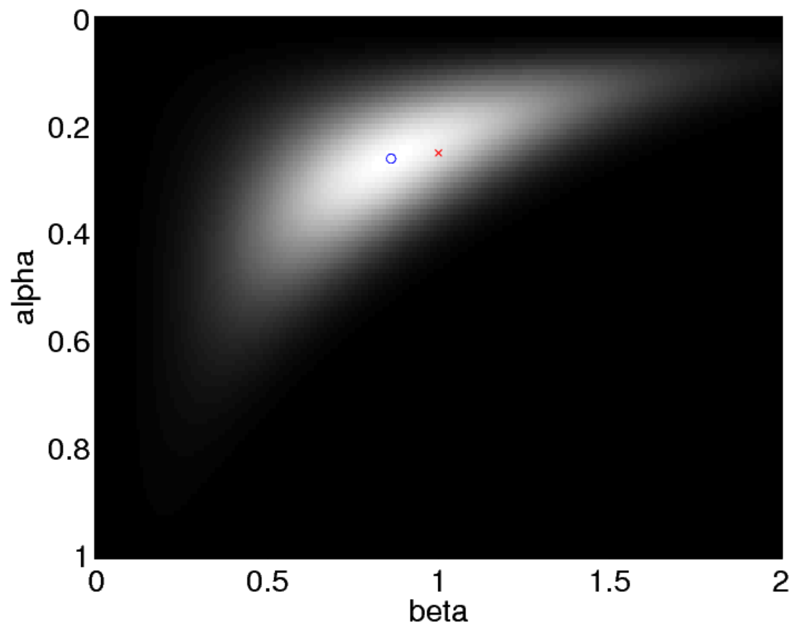
$\mathrm{argmax}_{\alpha,\beta}\, P(D \mid \alpha,\beta,M)$

(equivalently, can maximize the log likelihood)

Estimating model parameters

$$P(D \mid \alpha,\beta,M) = \prod P(c_t \mid \alpha,\beta,M)$$

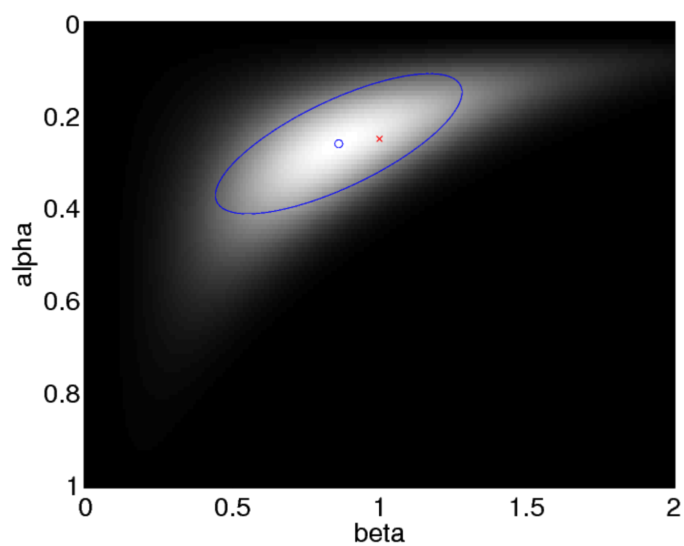
pragmatics:
fmincon, fminunc, fminsearch
minimize -LL
local minima: multiple starting points



simulated 1000 trials
α=0.25,β=1
recovered:
α=0.26,β=0.86

Error bars on estimates

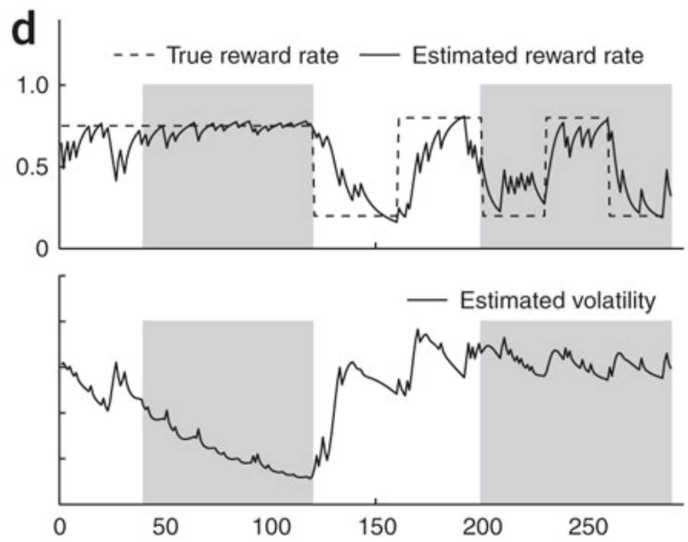
<u>intuition</u>: how fast likelihood is changing as parameters change

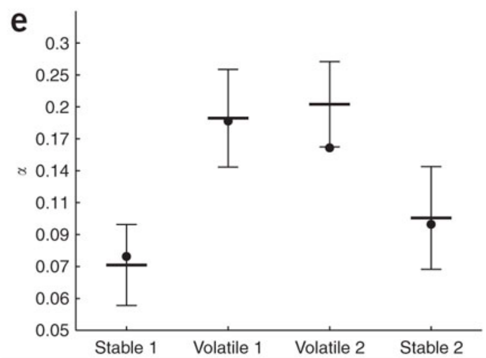


pragmatics: inverse Hessian (2nd derivative matrix; fmincon gives you this) of –LL estimates parameter covariance matrix
-error bars along the diagonal (sqrt), covariation off the diagonal
-can also look at variation in fits across subjects (we won't go into detail on this)

Example: do learning rates adapt to volatility?

Behrens et al. (2007)

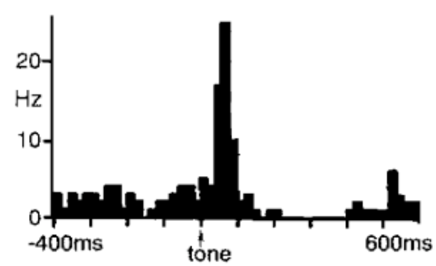




dots = Bayesian
bar + SE = humans (estimated from choice data)

Example: novelty bonuses

Wittmann et al (2008)

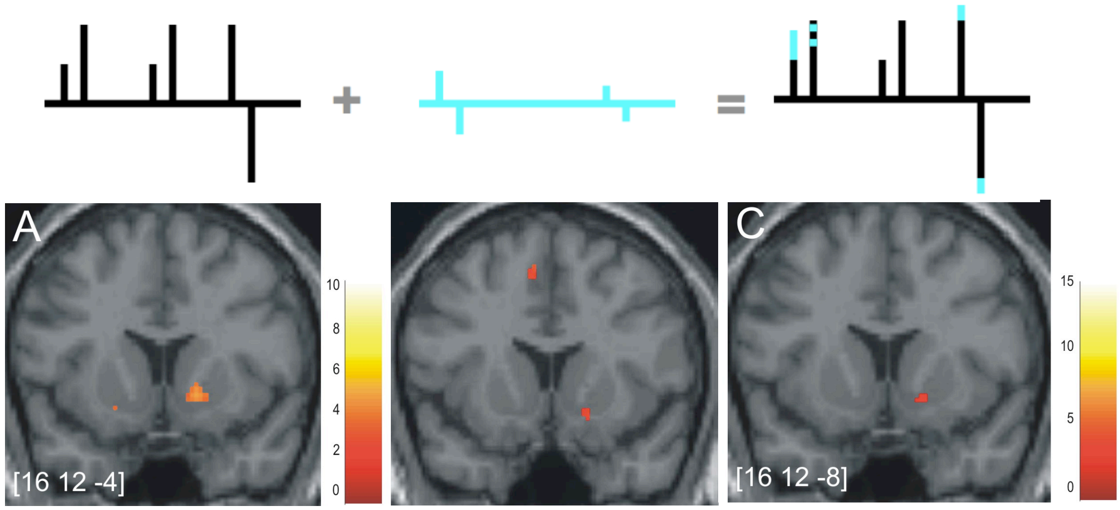


Horvitz et al. (1997)
dopamine neurons

Fit initial value separately for novel and preexposed images
initial value (preexposed) = £0.37
initial value (novel) = £0.41 (!!)

Example: novelty bonuses

Wittmann et al (2008)



Summary so far

- Learning models are detailed hypotheses about trial-by-trial overt and covert variables
- these can be tested against the brain
- help understand what different brain regions/networks are doing/computing
- a whole host of interesting results so far, but many questions still unanswered (relatively new method!)

- the models help us learn about the brain… can we also use the brain (or behavior) to learn about the models??

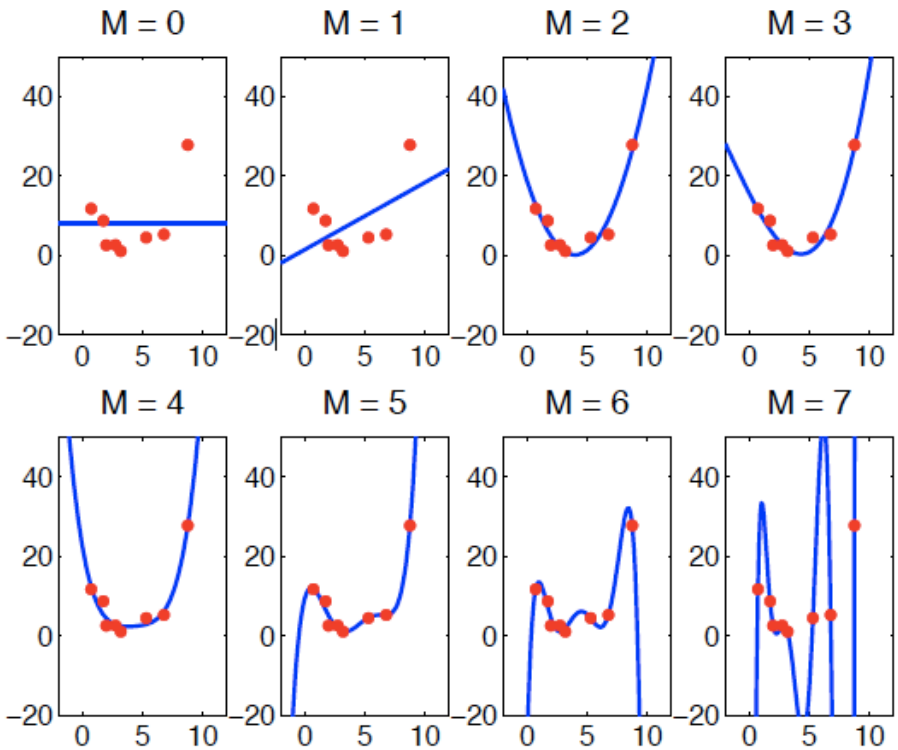
Act III:
What does my *data* tell me about my *model*?

Which model is best? Model comparison

- P(Model | Data) = ?
- comparing two models: $$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1) \cdot P(M_1)}{P(D|M_2) \cdot P(M_2)}$$

Bayes factor

Which model is best? Model comparison

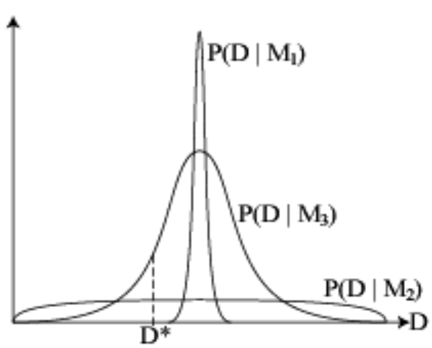


Which model is best? Model comparison

automatic Occam's razor: simple models tend to make precise predictions

can put in preference for simple models here, but don't need to...

- P(Model | Data) = ?
- comparing two models: $\frac{P(M_1|D)}{P(M_2|D)} = \boxed{\frac{P(D|M_1)}{P(D|M_2)}} \cdot \boxed{\frac{P(M_1)}{P(M_2)}}$
- "Pluralitas non est ponenda sine necessitate"
  Plurality should not be posited without necessity – William of Ockham (1349)
- we should go for the simplest model that explains the data



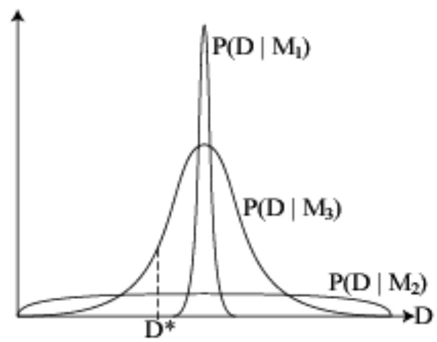
Which model is best? Model comparison

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1) \cdot P(M_1)}{P(D|M_2) \cdot P(M_2)}$$

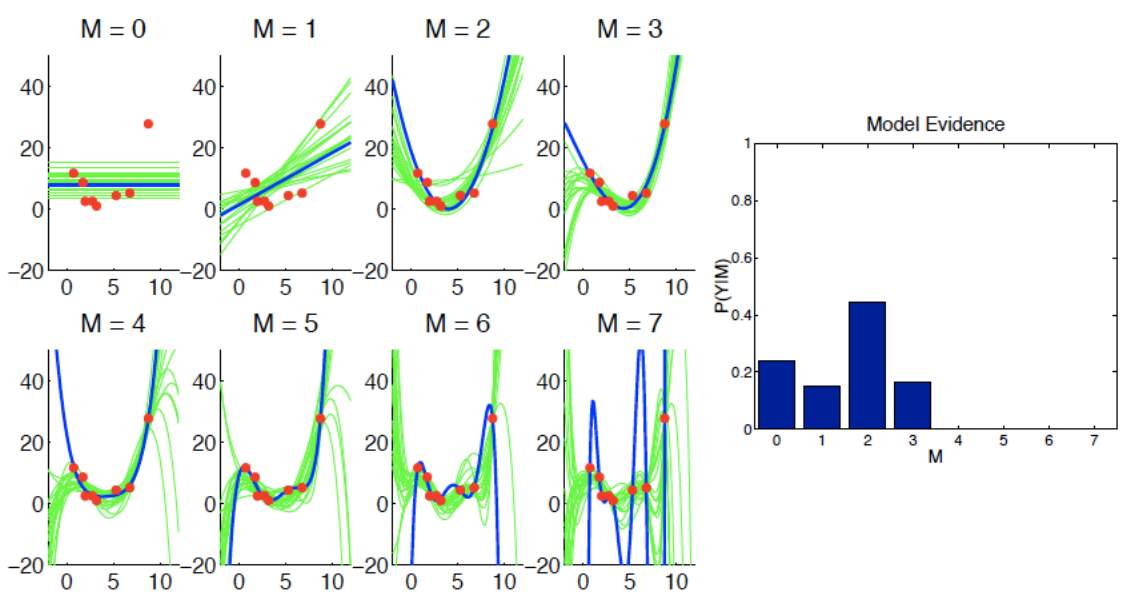
assuming uniform prior over models all we care about is P(D | M)

$$P(D|M) = \int d\theta P(D|M,\theta) \cdot P(\theta)$$

Bayesian evidence for model M (marginal likelihood)



Bayesian model comparison: Occam's Razor at work



Computing P(D | M)

$$P(D|M) = \int d\theta P(D|M,\theta) \cdot P(\theta)$$

- Integrating over all settings of the parameters is too hard...
- Approximate solutions:
  - sample posterior at many places to approximate integral and compute Bayes factor directly
  - Laplace approximation: make Gaussian approximation around MAP parameter estimate

Laplace approximation

$$P(D|M) = \int d\theta P(D|M,\theta) \cdot P(\theta)$$

- For large amounts of data (compared to # of parms d) the posterior is approximately Gaussian around the maximum a postriori (MAP) estimate $\hat{\theta}$

$$P(\theta|D,M) \approx 2\pi^{-\frac{d}{2}}|A|^{\frac{1}{2}} exp\left\{-\frac{1}{2}(\theta-\hat{\theta})^T A(\theta-\hat{\theta})\right\}$$

- and we also know that

$$P(D|M) = \frac{P(\theta,D|M)}{P(\theta|D,M)} = \frac{P(\theta|M)P(D|\theta,M)}{P(\theta|D,M)}$$

- so we can compute around the MAP estimate:

$$lnP(D|M) \approx lnP(\hat{\theta}|M) + lnP(D|\hat{\theta},M) + \frac{d}{2}ln(2\pi) - \frac{1}{2}ln|A|$$

- where -A is the Hessian matrix of lnP(θ | D,M)

$$A_{kl} = -\frac{\partial^2}{\partial\theta_{mk}\partial\theta_{ml}} lnP(\theta|D,M)|_{\hat{\theta}}$$

BIC approximation

$$lnP(D|M) \approx \boxed{lnP(\hat{\theta}|M)} + \boxed{lnP(D|\hat{\theta}, M)} + \boxed{\frac{d}{2}ln(2\pi)} - \boxed{\frac{1}{2}ln|A|}$$

prior on θ   data log likelihood   easy

- In the limit of LOTS of data ($N \rightarrow \infty$) A grows as NA_0 (for fixed A_0) so $\ln|A| = \ln|NA_0| = \ln N^d |A_0| = d\ln N + \ln|A_0|$.
- Retaining only terms that grow with N, we can approximate further:

$$lnP(D|M) \approx lnP(D|\hat{\theta}, M) - \frac{d}{2}ln(N)$$

(so, for each model we compute the log likelihood for the ML parameters and then add to that a penalty that depends on d (# of parameters), and then we compare the results between the models)

- Advantages: easy to compute; can use ML rather than MAP estimate
- Disadvantage: hard to determine d (only identifiable parameters) and N (only samples used to fit parameters; what if not same for diff parms?)

Non-Bayesian alternatives

- Likelihood ratio test: for nested models (one is a special case of the other; compares hypothesis H_1 to one where some parameters are fixed, H_0).
  - Statistical test on the likelihood differences: compare 2*difference in log likelihood (ML) to χ^2 statistic with df=#additional parameters
- AIC (Akaike's information criterion, 1974): measures goodness of a model based on bias and variance of the estimated model and measures of entropy.
  - Not statistical test, only ranks models.
  - Penalize log likelihood (ML) by adding # of parameters
- Fit models on training set and validate fit on hold-out set.
  - Problem: often hard to find two i.i.d. sets in a learning setting

Summary so far...

- Learning models are detailed hypotheses about trial-by-trial overt and covert variables
- trial-by-trial model fitting lets us test these hypotheses
- ...and compare alternatives
- special premium on detailed model fitting when considering learning data: non-stationary, can't use traditional averaging techniques
- a lot of leverage to pinpoint the neural correlates of learning and decision making in the brain

Additional reading

- Daw (2011) - Trial by trial data analysis using computational models
- Hare et al. (2008) - Dissociating the role of the orbitofrontal cortex and the striatum in the computation of goal values and prediction errors
- Niv (2009) - Reinforcement learning in the brain