Disk spindown problem

- Experts a set of n fixed timeouts : $\tau_1, \tau_2, \ldots, \tau_n$
- Master Alg.
  - maintains a set of weights : $s_1, s_2, \ldots, s_n$
  - predicts with weighted average
- Multiplicative update

$$s_{t+1,i} = \frac{s_{t,i}\ e^{-\eta \text{ energy usage of timeout i}}}{Z}$$

- Problem

Disk spindown problem

- ML 3: Mix in little bit of uniform vector

$$\begin{aligned} \boldsymbol{s}' &= \text{Multiplicative Update} \\ \boldsymbol{s} &= (1-\alpha)\,\boldsymbol{s}' + \alpha\,(\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}) \\ &\quad \text{where } \alpha \text{ is small} \end{aligned}$$

Disk spindown problem

- ML 3: Mix in little bit of uniform vector

$$\begin{aligned} \boldsymbol{s}' &= \text{Multiplicative Update} \\ \boldsymbol{s} &= (1-\alpha)\,\boldsymbol{s}' + \alpha\,(\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}) \\ & \quad \text{where } \alpha \text{ is small} \end{aligned}$$

- Nature 4: use mutation for same purpose

ML 4: sleeping

- Keep track of past average share vector $\boldsymbol{r}$

$$
\begin{aligned}
\boldsymbol{s}' &= \text{Multiplicative Update} \\
\boldsymbol{s} &= (1-\alpha)\,\boldsymbol{s}' + \alpha\,\boldsymbol{r}
\end{aligned}
$$

  - facilitates switch to previously useful vector
  - long-term memory

Sleeping interpretation - “Putting Bayes to sleep” [KAW]

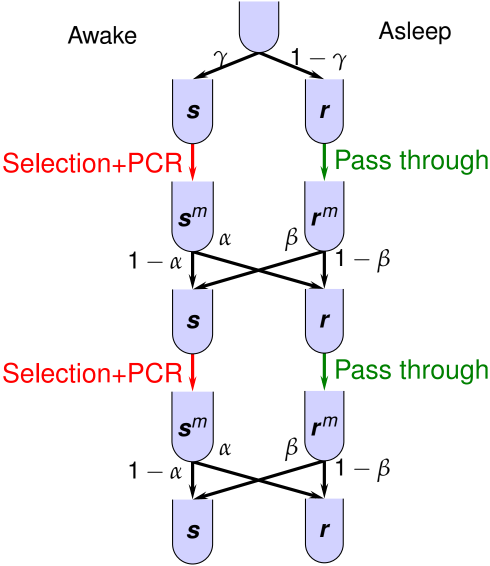
- Some models predict with predictive distribution
- Weights of those models fixed - Bayes rule is vacuous

Two-track in-vitro implementation of sleeping

Multiplicative update + small soup exchange gives long-term memory

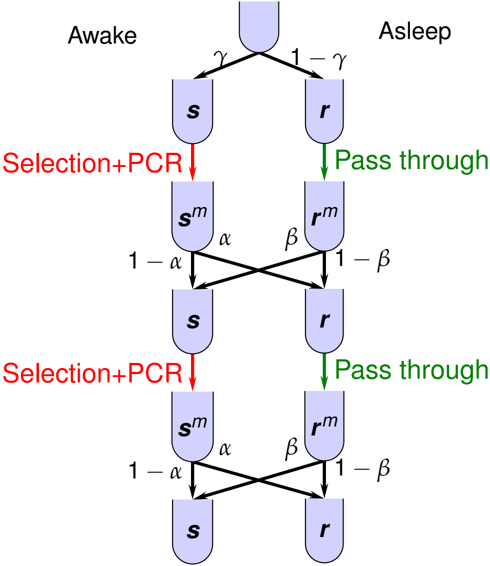
Two-track in-vitro implementation of sleeping

Multiplicative update + small soup exchange gives long-term memory



Two-track in-vitro implementation of sleeping

Multiplicative update + small soup exchange gives long-term memory



Initially:

$\boldsymbol{s} = \gamma$ "initial tube"

$\boldsymbol{r} = (1-\gamma)$ "initial tube"

$\boldsymbol{s}^m =$ "mult.update"$(\boldsymbol{s})$

$\boldsymbol{r}^m = \boldsymbol{r}$

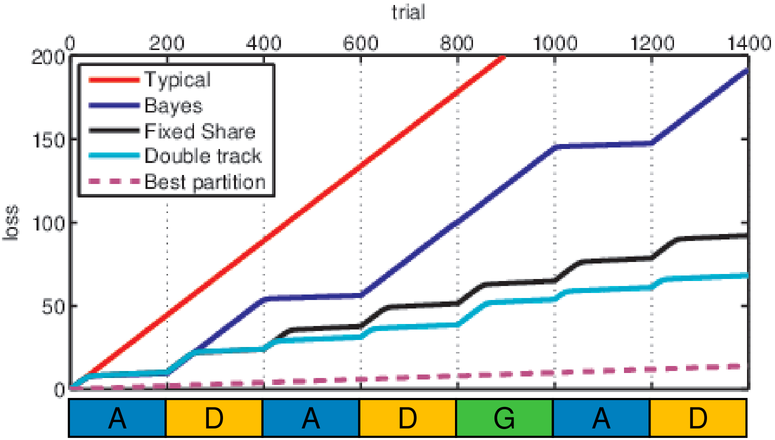
Soup exchange:

$\tilde{\boldsymbol{s}} = (1-\alpha)\boldsymbol{s}^m + \beta\boldsymbol{r}^m$

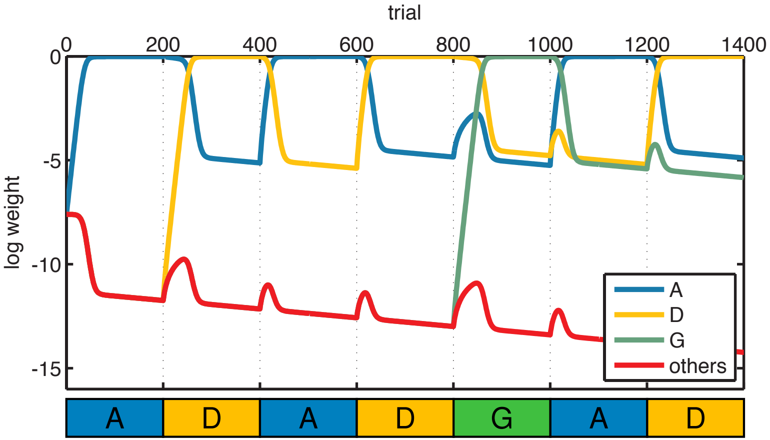
$\tilde{\boldsymbol{r}} = \alpha\boldsymbol{s}^m + (1-\beta)\boldsymbol{r}^m$

$\gamma = \frac{\beta}{\alpha+\beta}$

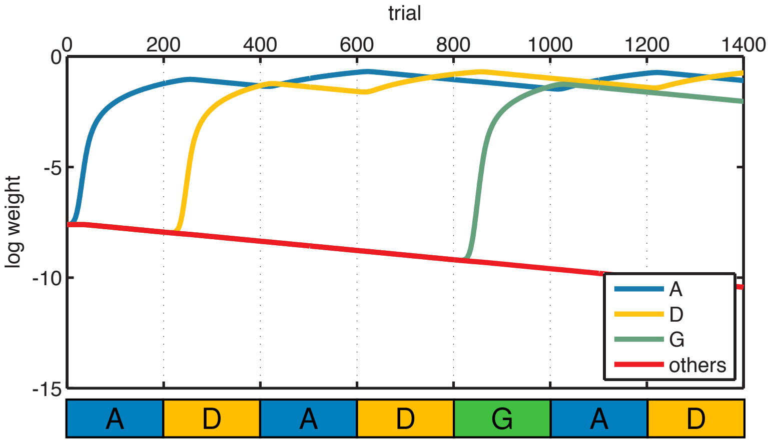
Total loss plots



Weights *s* on the awake track



Weights r on the sleeping track



Question

- How long-term memory realized in nature?
  - Junk DNA?
  - Sex?

ML methods

ML 1: Conservative update - learn multiple goals
ML 2: Upper bounding weights - ”
ML 3: Lower bounding weights - robust against change
ML 4: Mixing in past average/sleeping - longterm memory

Nature’s methods

Nature 1: Boundaries
Nature 2: Coupling
Nature 3: Super-predators
Nature 4: Mutations

Summary

- Multiplicative updates converge quickly - their blessing but wipe out diversity - their curse
- Changing conditions require reuse of previously learned knowledge/alternatives
- Diversity is a requirement for success
- A mechanism is need to ameliorate the curse
  - ML and Nature have different tricks