How
a beautiful and intriguing piece of technology provides
new insights in existing methods and
improves the state of the art in
expert tracking and
online multitask learning
We are trying to predict a sequence $y_1, y_2, \ldots$.

**Definition**

A **model** issues a prediction each round. We denote the prediction of model $m$ in round $t$ by $P(y_t | y_{<t}, m)$.

Say we have several models $m = 1, \ldots, M$. How to combine their predictions?
Bayesian answer

Place a **prior** $P(m)$ on models. Bayesian **predictive distribution**

$$P(y_t | y_{<t}) = \sum_{m=1}^{M} P(y_t | y_{<t}, m) P(m | y_{<t})$$

where **posterior distribution** is incrementally updated by

$$P(m | y_{\leq t}) = \frac{P(y_t | y_{<t}, m) P(m | y_{<t})}{P(y_t | y_{<t})}$$

Bayes is **fast**: predict in $O(M)$ time per round.

Bayes is **good**: regret w.r.t. model $m$ on data $y_{\leq T}$ bounded by

$$\sum_{t=1}^{T} (- \ln P(y_t | y_{<t}) + \ln P(y_t | y_{<t}, m)) \leq - \ln P(m).$$
Specialists

**Definition (FSSW97,CV09)**

A **specialist** may or **may not** issue a prediction. Prediction \( P(y_t | y_{<t}, m) \) only available for **awake** \( m \in W_t \).

Say we have several specialists \( m = 1, \ldots, M \).

How to combine their predictions?
Specialists

**Definition (FSSW97,CV09)**

A **specialist** may or **may not** issue a prediction. Prediction \( P(y_t|y_{<t}, m) \) only available for **awake** \( m \in W_t \).

Say we have several specialists \( m = 1, \ldots, M \). How to combine their predictions?

If we **imagine** a prediction for the sleeping specialists, we can use Bayes **Choice**: sleeping specialists predict with Bayesian predictive distribution

That **sounds** circular. Because it **is**. $%!#?
Bayes for specialists

Place a **prior** $P(m)$ on models. Bayesian **predictive distribution**

$$P(y_t|y< t) = \sum_{m \in W_t} P(y_t|y< t, m)P(m|y< t) + \sum_{m \notin W_t} P(y_t|y< t)P(m|y< t)$$
Bayes for specialists

Place a **prior** $P(m)$ on models. Bayesian **predictive distribution**

$$P(y_t|y_{<t}) = \sum_{m \in W_t} P(y_t|y_{<t}, m)P(m|y_{<t}) + \sum_{m \notin W_t} P(y_t|y_{<t})P(m|y_{<t})$$

has solution

$$P(y_t|y_{<t}) = \frac{\sum_{m \in W_t} P(y_t|y_{<t}, m)P(m|y_{<t})}{\sum_{m \in W_t} P(m|y_{<t})}.$$
Bayes for specialists

Place a **prior** \( P(m) \) on models. Bayesian **predictive distribution**

\[
P(y_t | y_{<t}) = \sum_{m \in W_t} P(y_t | y_{<t}, m) P(m | y_{<t}) + \sum_{m \notin W_t} P(y_t | y_{<t}) P(m | y_{<t})
\]

has solution

\[
P(y_t | y_{<t}) = \frac{\sum_{m \in W_t} P(y_t | y_{<t}, m) P(m | y_{<t})}{\sum_{m \in W_t} P(m | y_{<t})}.
\]

The **posterior distribution** is incrementally updated by

\[
P(m | y_{\leq t}) = \begin{cases} 
\frac{P(y_t | y_{<t}, m) P(m | y_{<t})}{P(y_t | y_{<t})} & \text{if } m \in W_t, \\
\frac{P(y_t | y_{<t}) P(m | y_{<t})}{P(y_t | y_{<t})} = P(m | y_{<t}) & \text{if } m \notin W_t.
\end{cases}
\]

Bayes is **fast**: predict in \( \mathcal{O}(M) \) time per round.

Bayes is **good**: regret w.r.t. model \( m \) on data \( y_{\leq T} \) bounded by

\[
\sum_{t : 1 \leq t \leq T \text{ and } m \in W_t} \left( -\ln P(y_t | y_{<t}) + \ln P(y_t | y_{<t}, m) \right) \leq -\ln P(m).
\]
The specialists trick

Specialists just a curiosity?

Start with input models
Create many derived virtual specialists
Control how they predict
Control when they are awake
Run Bayes on all these specialists
Carefully choose prior
Low regret w.r.t. class of intricate combinations of input models.
Fast execution. Bayes on specialists
The specialists trick

Specialists just a curiosity?

Specialists trick:
  - Start with input models
  - Create many derived virtual specialists
    - Control how they predict
    - Control when they are awake
  - Run Bayes on all these specialists
  - Carefully choose prior
    - Low regret w.r.t. class of intricate combinations of input models.
    - Fast execution. Bayes on specialists collapses
Application 1: Freund’s Problem

In practice:
- Large number $M$ of models
- Some are good some of the time
- Most are bad all the time

We do not know in advance which models will be useful when.

**Goal:** compare favourably on data $y_{\leq T}$ with the best alternation of $N \ll M$ models with $B \ll T$ blocks.

Bayesian reflex: average all partition models. **NP hard.**
Application 1: sleeping solution

Create **partition specialists**

Bayes is **fast**: $O(M)$ time per trial, $O(M)$ space.
Bayes is **good**: regret close to information-theoretic lower bound.
Application 1: sleeping solution

Create partition specialists

Bayes is **fast**: $O(M)$ time per trial, $O(M)$ space
Bayes is **good**: regret close to information-theoretic lower bound

- Bayesian interpretation for Mixing Past Posteriors
- Faster algorithm
- Slightly improved regret bound
- Explain mysterious factor 2
Application 2: online multitask learning

In practice:
- Large number $M$ of models
- Data $y_1, y_2, \ldots$ from $K$ interleaved tasks
- Observe task label $\kappa_1, \kappa_2, \ldots$ before prediction

We do not know in advance which tasks are similar.

**Goal:** Compare favourably to best clustering of task with $N \ll K$ cells

Bayesian reflex: average all cluster models. NP hard.
Application 2: sleeping solution

Create cluster specialists

Bayes is fast: $O(M)$ time per trial, $O(MK)$ space
Bayes is good: regret close to information-theoretic lower bound

- Intriguing algorithm
- Regret independent of task switch count
Conclusion

The **specialists trick**

- NP-hard offline problem
- Ditch Bayes on *coordinated* comparator models
- Instead create *uncoordinated* virtual specialists
- Craft prior so that
  - Bayes collapses (efficient emulation)
  - Small regret
The *specialists trick*

- NP-hard offline problem
- Ditch Bayes on *coordinated* comparator models
- Instead create *uncoordinated* virtual specialists
- Craft prior so that
  - Bayes collapses (efficient emulation)
  - Small regret

Thank you!