Thompson Sampling in Switching Environments

with Bayesian Online Change Point Detection

The University of Manchester

Joseph Mellor mellorj@cs.man.ac.uk

Jonathan Shapiro jls@cs.man.ac.uk

Goal

Design and evaluate a Thompson Sampling algorithm for switching environments

Motivation

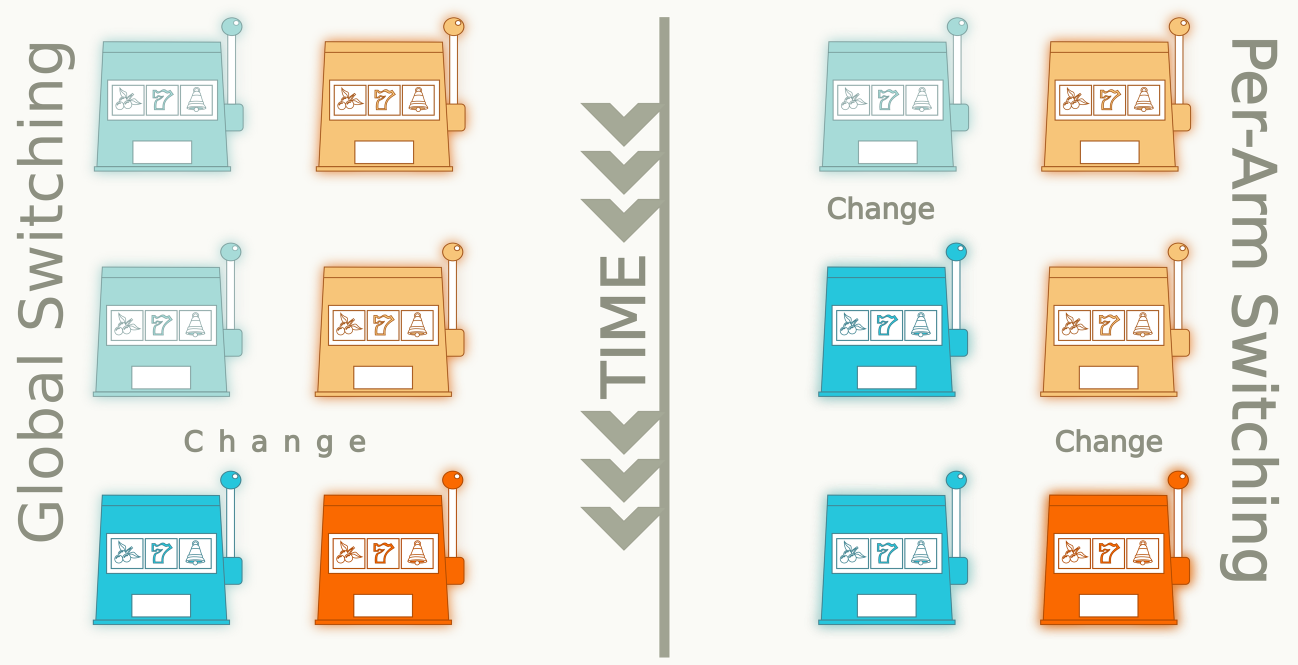
Many problems require making a decision under uncertainty while continually improving future decisions based on the outcome. Some examples might be in clinical trials, web advertising, and finance.

Multi-armed bandits are a well studied simple model for such problems. Thompson Sampling has been shown empirically to perform well and achieve the lower bound on regret for the stationary Bernoulli Multi-armed bandit [5].

Real world scenarios are rarely stationary. Some environments change abruptly such as scenarios based on,

- Financial data (e.g. stock prices)
- Structural networks (e.g. congestion/failure)

Switching Environment



The model for which we design our bandit algorithm is as follows. There are N arms. At time t arm i will return a reward $c_{i,t} \in \{0,1\}$ from a Bernoulli trial with the expectation $\mu_{i,t}$. We imagine a constant switching rate γ , such that,

$$\mu_{i,t} = \begin{cases} \mu_{i,t-1} & \text{with probability, } 1-\gamma \\ \mu_{\text{new}} \sim \text{uniform U(0,1)} & \text{with probability, } \gamma \end{cases}$$

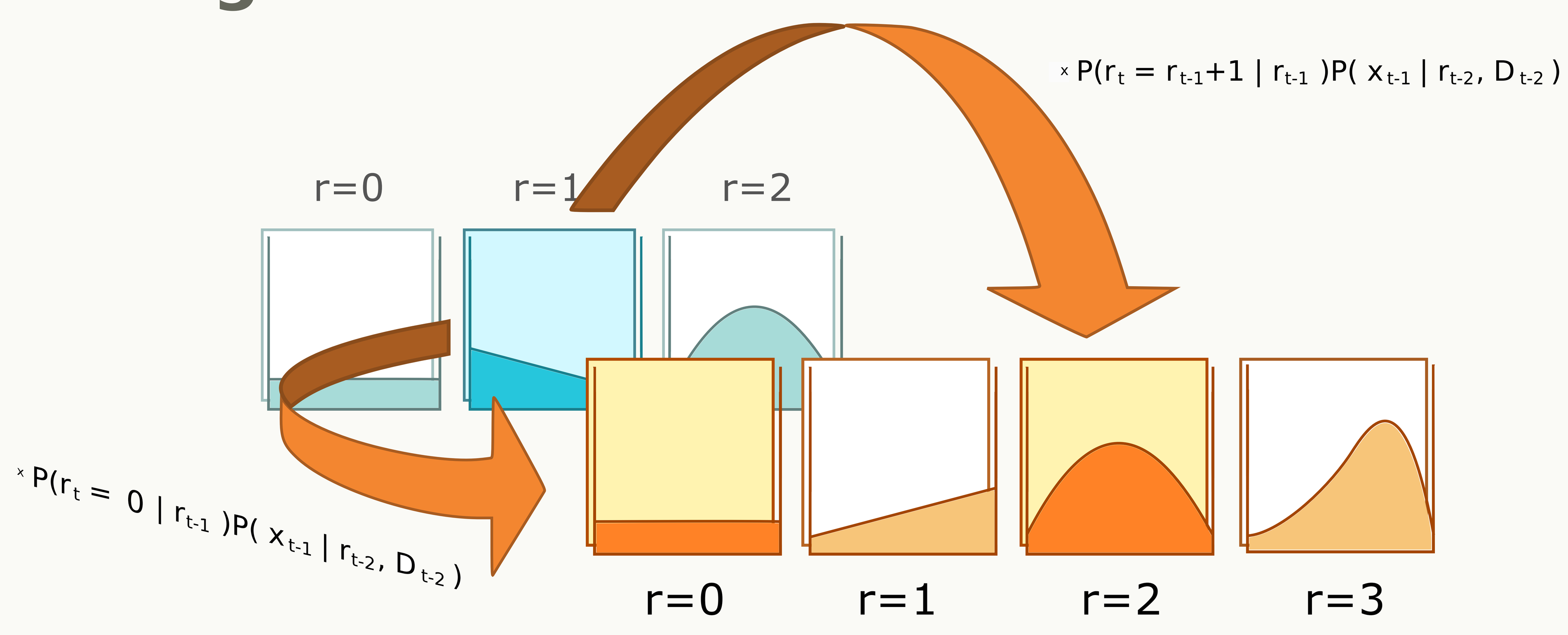
Thompson Sampling

Thompson Sampling is a probability matching algorithm. We pull an arm with the probability that the arm is the best arm. This probability is

$$P(a_i = a^*) = \int_\theta I(a_i = a^* | \theta) P(\theta) d\theta$$

where θ is a model of our arms, a^* is the optimal arm and a_i is the ith arm. $I(x)$ is the indicator function. The strategy is reduced to sampling from $P(\theta)$ and picking the arm that is maximal for the sample.

Change Point Detection



We detect change points by inferring the distribution of the runlength, r_t, the time elapsed since the last change point, from the past history, D_t, of arm chosen and reward received, x_t. Inference is done via a message-passing algorithm [3,4]

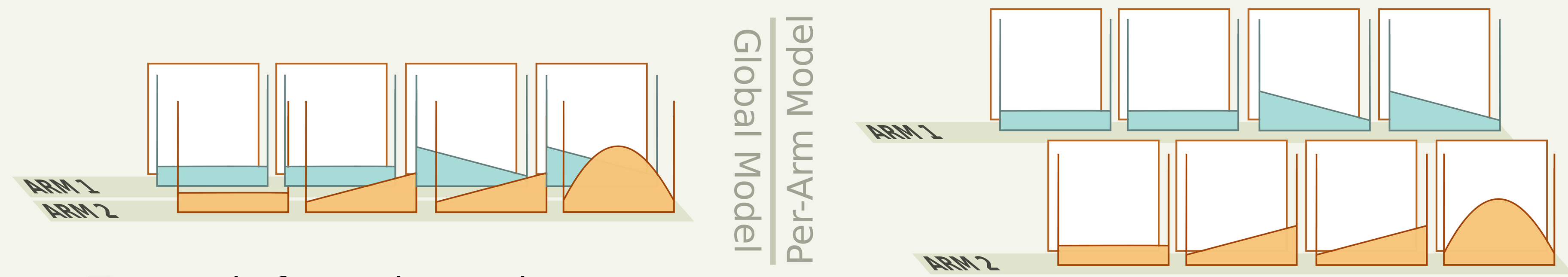
$$P(r_t, x_{t-1}, D_{t-2}) = \sum_{r_{t-1}} \underbrace{P(r_t|r_{t-1})}_{\text{switching rate}} \underbrace{P(x_{t-1}|r_{t-1}, D_{t-2})}_{\text{reward likelihood}} P(r_{t-1}, x_{t-2}, D_{t-3})$$

This is efficient since there are only two possibilities, either the runlength increases by one ($r_t = r_{t-1} + 1$) or a change point occurs and the runlength becomes zero ($r_t = 0$).

Combining Thompson Sampling and Change Point Detection

The model contains

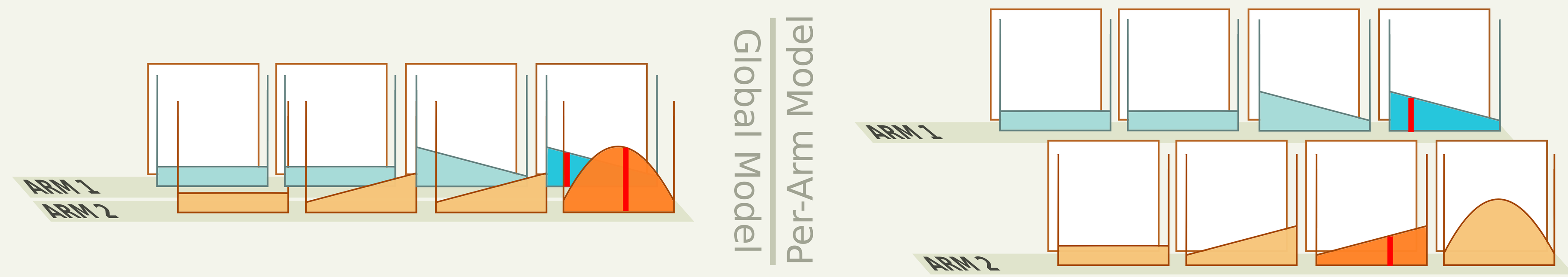
- A Runlength Distribution (1 for Global, N for Per-Arm)
- Beta hyperparameters for each (arm,runlength) pair



To sample for each arm, j, we

- Sample a runlength r(j) from the runlength distribution(s)
- Sample θ(j) from the Beta distribution for j associated with r(j)

We then pick the arm j with largest θ(j).



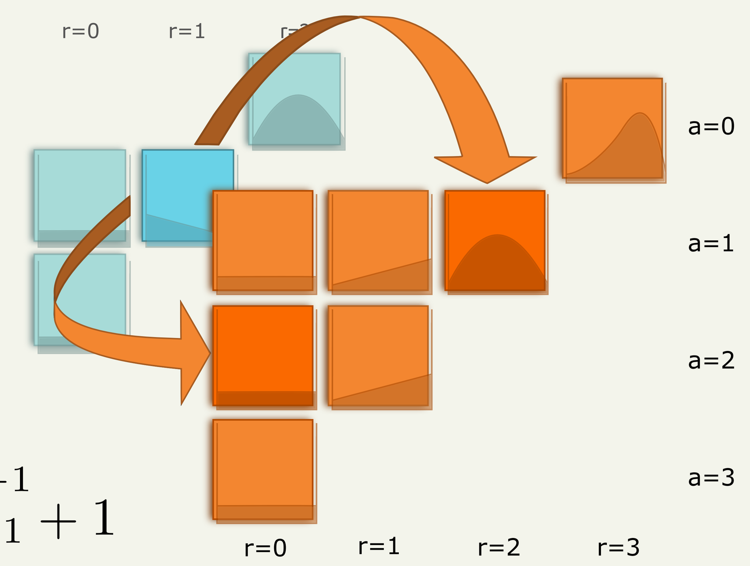
We call these algorithms **Global Changepoint Thompson Sampling** (Global CTS) and **Per-Arm Changepoint Thompson Sampling** (Per-Arm CTS).

Limiting Space Requirements

The runlength grows with time. We use Stratified Optimal Resampling, a particle filter resampling technique to limit the space requirements of the algorithm [4].

Switch Rate Inference

- Infer the joint distribution of the runlength and the number of change points that have occurred [1].
- Number of change points, a_t, tells us the switch rate
- Still only two possibilities $\begin{matrix} a_t = a_{t-1} + 1 \\ r_t = 0 \end{matrix}$ or $\begin{matrix} a_t = a_{t-1} \\ r_t = r_{t-1} + 1 \end{matrix}$
- Now scales quadratically rather than linearly with time



We call these algorithms **Non-parametric Global/Per-Arm Changepoint Thompson Sampling** (NP Global/Per-Arm CTS).

Experiments

	Global	Per-Arm	PASCAL *	ForEx **	Yahoo!
Global CTS	**5.9±0.5**	13.8±1.4	67.2±4.1	351.9±28	0.489±0.035
Per-Arm CTS	12.1±0.1	**13.0±0.1**	39.6±8.9	370.4±14	0.522±0.028
NP Global CTS	6.7±0.1	13.8±0.2	67.9±4.3	**348.2±14**	0.490±0.029
NP Per-Arm CTS	29.4±1.0	30.8±0.8	93.2±6.5	353.5±14	0.590±0.018
Global CTS 2	30.5±1.1	37.9±1.0	35.8±3.2	358.0±14	**0.443±0.031**
Per-Arm CTS 2	49.6±1.7	67.1±1.2	37.4±4.1	380.9±13	0.505±0.028
DiscountedUCB	15.5±1.9	16.8±2.0	**35.7±3.8**	606.3±32	0.563±0.018
UCB	178.3±58	175.1±53	161.2±9.1	613.9±35	0.526±0.040
Random	333.1±15	336.4±13	321.3±11	623.3±29	0.800±0.001

- Algorithms perform well in environments matching their generative model (Global, Per-Arm).
- Can be made competitive with other algorithms for other non-stationary environments
- Inferring the switch rate can often cause degradation in performance

* PASCAL EvE Challenge 2006
** Foreign Exchange data

References

[1] **Robert C. Wilson, Matthew R. Nassar, and Joshua I. Gold.** Bayesian online learning of the hazard rate in change-point problems. Neural Comput., 22 (9):2452{2476, 2010. ISSN 0899-7667. doi: 10.1162/NECO a 00007. URL http://dx.doi.org/10.1162/NECO_a_00007.

[2] **Yahoo!** Yahoo! Webscope Dataset ydata-frontpage-todaymodule-clicks-v1 0. http://labs.yahoo.com/Academic_Relations, 2011.

[3] **Ryan Prescott Adams and David J.C. MacKay.** Bayesian online changepoint detection. Cambridge, UK, 2007.

[4] **Paul Fearnhead and Zhen Liu.** On-line inference for multiple change points problems. Journal of the Royal Statistical Society B, 69:589–605, 2007.

[5] **Emilie Kaufmann, Nathaniel Korda, and Rémi Munos.** Thompson sampling: An optimal finite time analysis. CoRR, abs/1205.4217, 2012.

Many thanks to the EPSRC for funding this research