Dynamic Programming & Reinforcement Learning

Subject 11
Bayesian dynamic programming

Ger Koole

Motivating example

- Suppose you have the choice between 2 medications
- No information about "success" probabilities: we have to learn them
  - only information: series of 0/1, e.g. 001001100
- Objective: max expected discounted # of cured patients
- No states, just choice between different actions: stateless bandits
- Crucial:
  - how to learn the value = expected reward of actions
  - how to make sure we find the right arm while not investing too much in suboptimal arms = exploration-exploitation trade-off

Stateless bandits

- Suppose you do not have a model?
- Can you learn the value "on the fly"?
  - Online, as compared to offline when you have a model
  - You do not have transition probabilities, only realised transitions
- Simple rules: last reward, average of rewards

Stateless bandits

- Historical rewards $r_1, \ldots, r_t$ of an arm, realizations of random variable R_i
- Prediction methods:
  - Average: $Q_t = (r_1 + \cdots + r_t)/t$
  - LLN (law of large numbers): if R_i i.i.d. then $Q_t \rightarrow ER_i$ (i.i.d. = independent and identically distributed)
  - Note:

$$Q_t = (r_1 + \cdots + r_t)/t = r_t/t + (r_1 + \cdots + r_{t-1})/t = r_t/t + (t-1)/tQ_{t-1} = r_t/t + (1-1/t)Q_{t-1}$$

  - More general and simpler to implement:

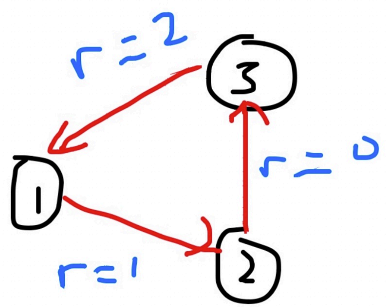
$$Q_t = \gamma_t r_t + (1-\gamma_t)Q_{t-1}$$

  - Theorem: if R_i i.i.d., $\Sigma_t \gamma_t = \infty$ and $\Sigma_t \gamma_t^2 < \infty$ then $Q_t \rightarrow ER_i$

Stateless bandits

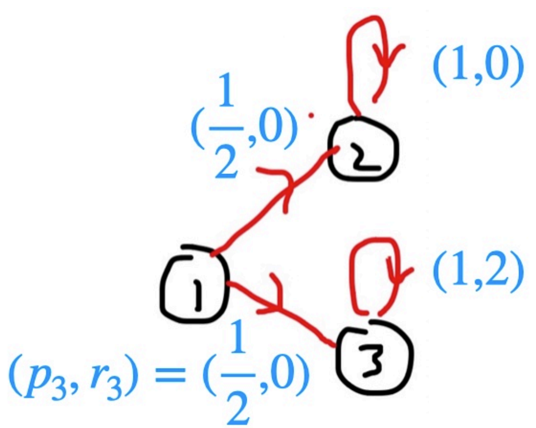
- Prediction methods
  - we had: $Q_t = \gamma_t r_t + (1 - \gamma_t) Q_{t-1}$
  - disadvantages: converges slowly, assumes stationarity
  - common choice: $\gamma_t = \gamma$
    - equivalent to exponential smoothing & discounting
    - same reason as exp smoothing: old values are less valuable
    - Q-learning with $|\mathscr{X}| = 1$

Examples




```
r=rep(c(1,0,2),1000)
Q=rep(0,3000)
gamma=1/(1:3000)
for(t in 2:3000)Q[t]=gamma[t]*r[t]+(1-gamma[t])*Q[t-1]
head(Q);tail(Q)
```

```
[1] 0.0000000 0.0000000 0.6666667 0.7500000 0.6000000 0.8333333
[1] 0.9996661 0.9993324 0.9996663 0.9996664 0.9993331 0.9996667
```




```
r=c(1,rep(sample(c(0,2),1),2999))
Q=rep(0,3000)
gamma=1/(1:3000)
for(t in 2:3000)Q[t]=gamma[t]*r[t]+(1-gamma[t])*Q[t-1]
head(Q);tail(Q)
```

sample with: r[3,]=(0,0,0,…)

```
[1] 0 0 0 0 0 0
[1] 0 0 0 0 0 0
```

sample with: r[3,]=(0,2,2,…)

```
[1] 0.000000 1.000000 1.333333 1.500000 1.600000 1.666667
[1] 1.999332 1.999332 1.999333 1.999333 1.999333 1.999333
```

(if you know the states: Q-learning)

Exploration policies

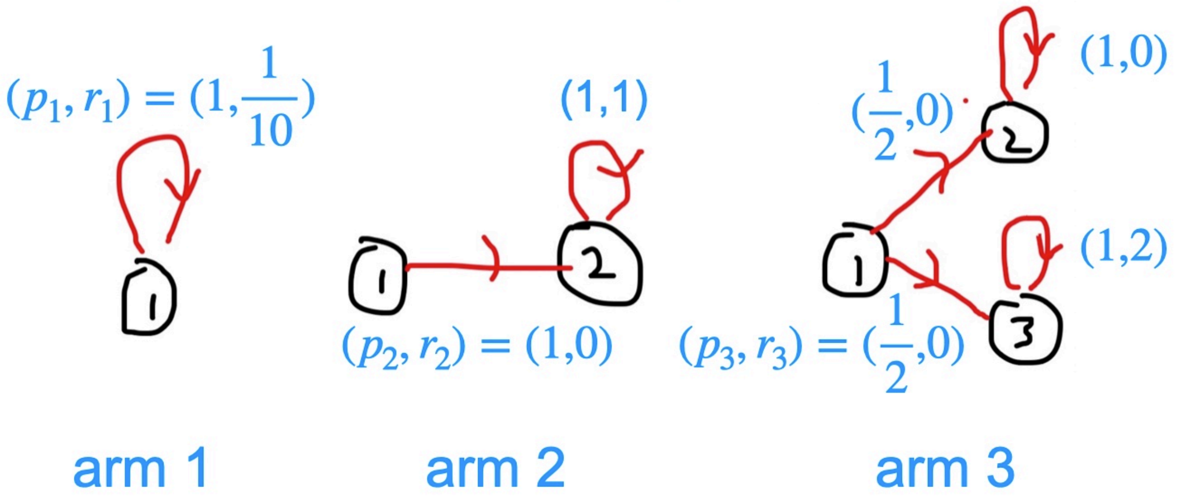
- Exploitation: choose arm $\text{argmax}_i \, Q_{N(i)}(i)$
- Greedy policy: always exploit
  - no guarantee to find optimal arm
- ε-greedy: with probability ε: take random arm
  - guaranteed to find optimal arm
  - wastes ε no matter how bad other arms are
- Greedy with optimistic initial values: start with Q_0 high
  - Initially enough exploration
  - How high? Too high = too much exploration, too low = might miss best arm
- Wanted: methods that take difference in prediction and variability into account: UCB

UCB

- UCB = upper confidence bound
- in general: CI = [av - c σ / √ t, av + c σ / √ t]
- because of recursion Q_t σ not known
- select arm $\text{argmax}_a \{ Q_t(a) + c' \sqrt{(\log t / N_t(a))} \}$
  where $N_t(a)$ = number of times you played arm a
- based on Hoeffding's inequality:

$$P(EX > (\Sigma_1^t X_i)/t + C) < \exp(-2tC^2)$$

Example optimistic values




```r
r=matrix(0,nrow=3,ncol=3000);Q=r;Q[,1]=c(2,2,2) # optimistic values
r[1,]=rep(0.1,3000);r[2,]=c(0,rep(1,2999));r[3,]=c(0,rep(sample(c(0,2),1),2999))
N=c(1,1,1)
gamma=1/(1:3000)
for(t in 2:3000){
  a=which.max(c(Q[1,N[1]],Q[2,N[2]],Q[3,N[3]]))
  Q[a,N[a]+1]=gamma[t]*r[a,N[a]]+(1-gamma[t])*Q[a,N[a]]
  N[a]=N[a]+1
  }
Q[1:3,1:5];N
```

reward arm 3: (0,0,0,…) or (0,2,2,…), both with pr. 0.5

sample with: r[3,]=(0,0,0,…)

time t		[,1]	[,2]	[,3]	[,4]	[,5]
value Q[1,t]	[1,]	2	1.050000	1.026250	1.014375	1.008069
value Q[2,t]	[2,]	2	1.333333	1.277778	1.238095	1.208333
value Q[3,t]	[3,]	2	1.500000	1.200000	1.080000	1.036800
	[1]	10	2981	11		

most chosen action: 2

and (0,2,2,…)

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	2	1.050000	0.0	0.000000	0.000000
[2,]	2	1.333333	0.0	0.000000	0.000000
[3,]	2	1.500000	1.6	1.666667	1.714286
[1]	2	2	2998		

most chosen action: 3

Conclusion

- Exploration heuristics simpler than GI but suboptimal
- Vincent: extend learning heuristics to state spaces
- Ger: extend state space to learning problems

Back to example

- Choice between 2 medications
- No information about “success” probabilities: stateless bandits
- Suppose you have the choice between 2 medications
  - blue: success probability = 0.4
  - red: harmless but with unknown success probability

Poll!

- Again but
  - blue: success probability = 0.6
  - red: unknown success probability

Poll!

Known versus unknown success probability

- Known:

```
> sample(c(0,1),50,TRUE,c(0.5,0.5))
 [1] 1 1 1 0 0 0 0 1 0 0 0 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1
> sample(c(0,1),50,TRUE,c(0.5,0.5))
 [1] 1 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1 0 1 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 1
> sample(c(0,1),50,TRUE,c(0.5,0.5))
 [1] 0 0 1 0 1 0 1 0 0 0 1 0 1 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1 1
```

- Unknown, all probabilities equally likely = uniform distribution on [0,1]

```
> gamma=runif(1,0,1);sample(c(0,1),50,TRUE,c(1-gamma,gamma))
 [1] 0 1 1 1 0 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 0 0 0 1 0
> gamma=runif(1,0,1);sample(c(0,1),50,TRUE,c(1-gamma,gamma))
 [1] 0 1 1 1 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 0
> gamma=runif(1,0,1);sample(c(0,1),50,TRUE,c(1-gamma,gamma))
 [1] 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1
```

RL heuristics

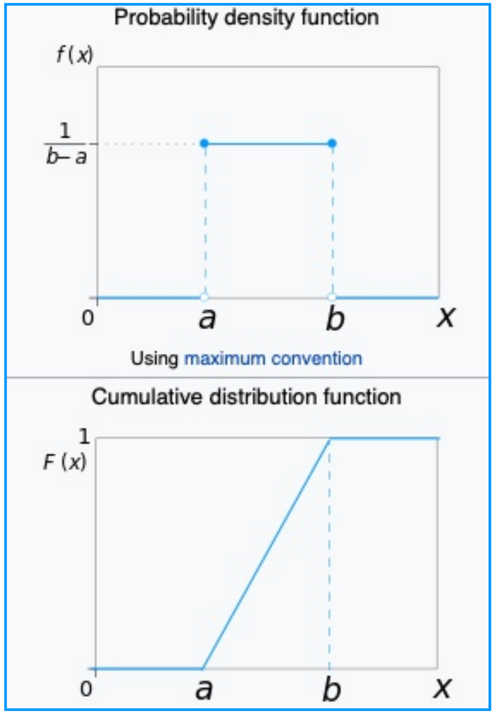
- Up to now:
  - value of bandit = $Q_t = \gamma_t r_t + (1 - \gamma_t) Q_{t-1}$ with $r_t = 0$ (not cured) or 1 (cured)
  - Exploration/exploitation strategy such as ε-greedy & optimistic values (e.g., start with $Q_0 = 1$)
  - Demo: https://gdmarmerola.github.io/ts-for-bernoulli-bandit/
- Can this be done smarter?

Framework for a medication

Bayesianism [edit]

Bayesian probability is the name given to several related interpretations of probability as an amount of epistemic confidence – the strength of beliefs, hypotheses etc. – rather than a frequency. This allows the application of probability to all sorts of propositions rather than just ones that come with a reference class. "Bayesian" has been used in this sense since about 1950. Since its rebirth in the 1950s, advancements in computing technology have allowed scientists from many disciplines to pair traditional Bayesian statistics with random walk techniques. The use of the Bayes theorem has been extended in science and in other fields.[14]

- Not certain of success probability = distribution on success probabilities
- Observation (0/1) gives information on distribution ⇒ updated distribution
  - Bayesian framework
  - prior → observation → updated prior = posterior
- Here: prior U is Uniform on [0,1], $f_U(x) = 1$ for $x \in [0,1]$, 0 otherwise



(from wikipedia)

Posterior

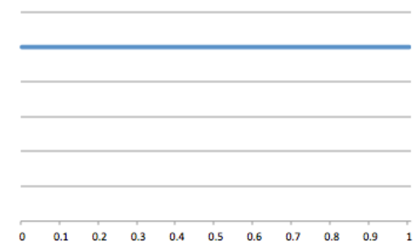
- $U \sim$ Uniform$[0,1]$, $X \sim$ Bernoulli(U)
  - Thus X is 0 or 1 with probability U
- Posterior depends on outcome X
- posterior for outcome 1 $= F_{U|X=1}(p) =$

$$P(U \leq p \,|\, X = 1) = \frac{P(U \leq p \;\&\; X = 1)}{P(X = 1)} =$$

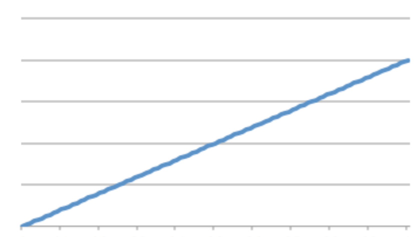
$$\frac{\int_0^p P(X = 1 \,|\, U = q) f_U(q) dq}{\int_0^1 P(X = 1 \,|\, U = q) f_U(q) dq} = \frac{\int_0^p q dq}{\int_0^1 q dq} = \frac{\frac{p^2}{2}}{\frac{1}{2}} = p^2$$

- Thus: density of posterior for outcome 1

$$f_{U|X=1}(p) = \frac{d}{dp} F_{U|X=1}(p) = 2p$$

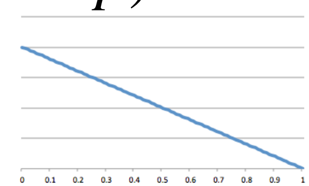


success



Posterior

- Similarly: $f_{U|X=0}(p) = \dfrac{d}{dp} F_{U|X=0}(p) = \dfrac{d}{dp} - (1-p)^2 = 2(1-p)$

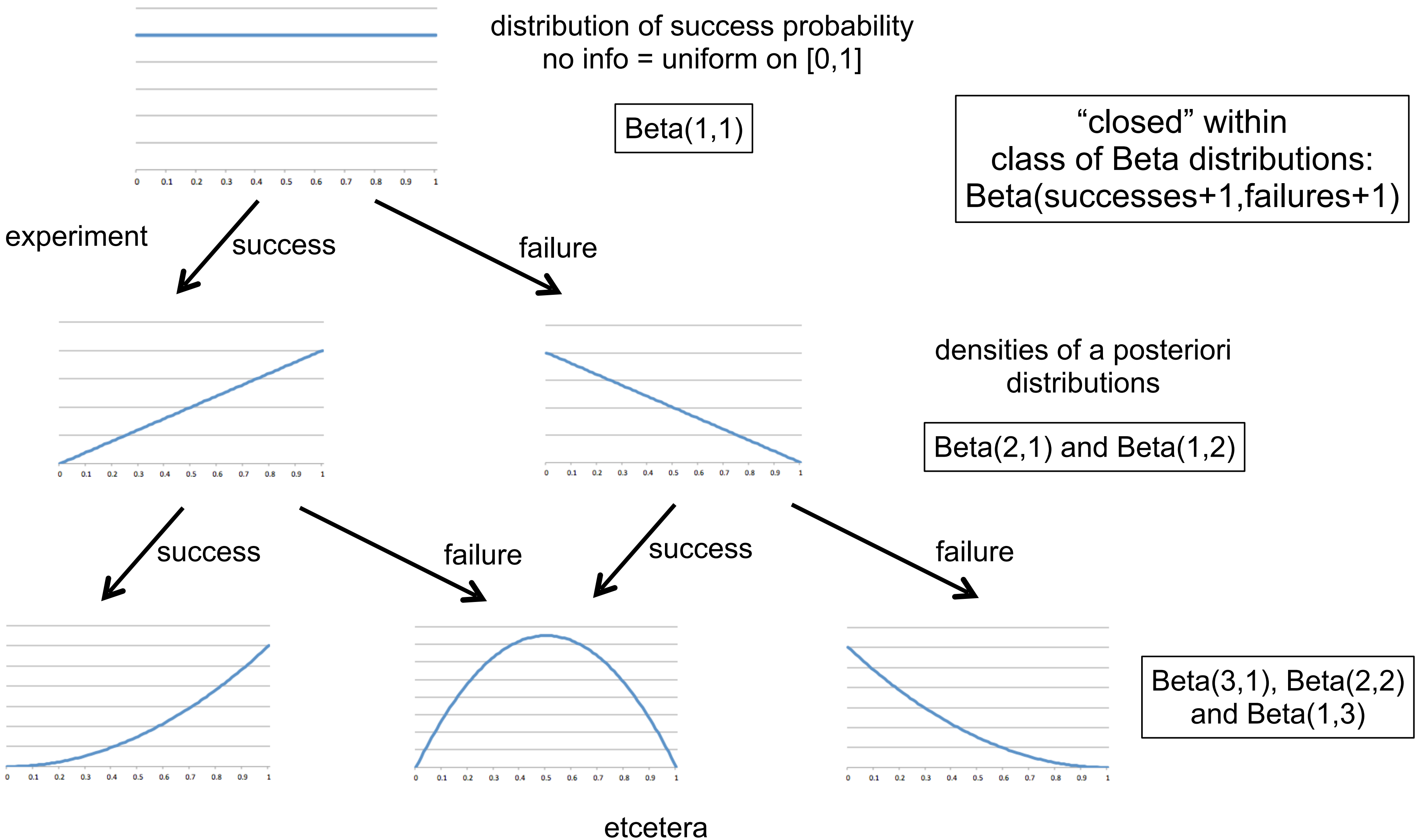


- Another arrival
  - E.g., $X_1 = 1, X_2 = 0$, define $U_1 = U \mid X_1 = 1$
  - $F_{U_1|X_2=0}(p) = P(U_1 \leq p \mid X_2 = 0) = \dfrac{P(U_1 \leq p \;\&\; X_2 = 0)}{P(X_2 = 0)} =$

$$\frac{\int_0^p P(X_2 = 0 \mid U_1 = q) f_{U_1}(q)\,dq}{\int_0^1 P(X_2 = 0 \mid U_1 = q) f_{U_1}(q)\,dq} = \frac{\int_0^p (1-q)2q\,dq}{\int_0^1 (1-q)2q\,dq} = \frac{p^2 - \frac{2}{3}p^3}{\frac{1}{3}} = 3p^2 - 2p^3$$

  - $f_{U_1|X_2=0}(p) = \dfrac{d}{dp} F_{U_1|X_2=0}(p) = 6p(1-p)$
- Etcetera… density of $U \mid k$ successes, l failures is $(k+l+1)!p^k(1-p)^l$
- Beta distributions (with parameters $k+1$ and $l+1$)

Beta distributions



Exploration policies

- Demo: https://gdmarmerola.github.io/ts-for-bernoulli-bandit/
- How to utilise posterior distribution?
- New exploration strategy: Thompson sampling
  - sample from “belief” = posterior at each epoch
  - take action according to argmax reward
  - very good because takes variability into account
- Example
  - blue: success probability = 0.6
  - red: Uniform[0,1] prior
  - What does Thompson sampling do? Poll!
- Demo
- Can we do even better?

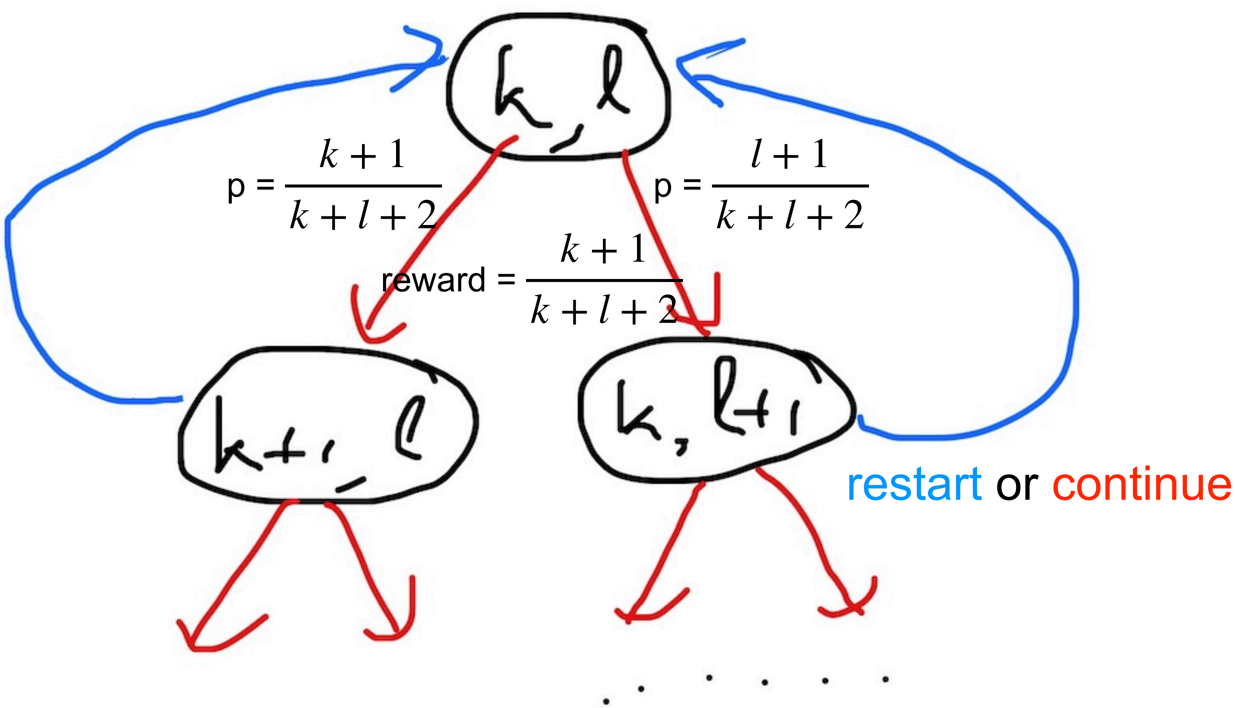
Exploration policies

- How to utilise posterior distribution?
- By using (k, l) as state each medication becomes a model-based bandit
  - The Gittins index can be computed
  - Which gives the optimal policy

Gittins index

$$P(X=1) = EB = \frac{k+1}{k+l+2} \text{ with } B \sim \text{Beta}(k+1, l+1) \text{ prior}$$

$= (k, l)$ successes/failures = state



Numerical example

- 2 uniformly distributed BBs, $\beta = 0.99$
- Note that $0.95^{1000} \approx 5 \times 10^{-23}$, $0.99^{1000} \approx 5 \times 10^{-5}$
- 10K runs of length 1000:

full info	66.60153	prior knowledge
Bayesian	64.97616	uniform[0,1] prior belief
Thompson	57.23021	
greedy	63.88666	
optimistic Q-learning	64.02162	initial values 1
eps-greedy Q-learning	54.25388	$\varepsilon = 10\,\%$
eps-first	52.23387	200 times random then greedy

- Run time: >3h on 2017 macbook pro

RCTs vs bandits

- Framework of Bernoulli bandits (BBs) applicable to many other areas such as advertising and web-page design
- Dominant framework in health care: randomised controlled trials (RCTs)
  - completely separates exploration and exploitation
    - exploration with 200 subjects, 100 in control group
    - statistical test to determine best treatment
    - then exploitation
  - Advantages RCTs: simple, easy to verify if used in the right way
  - Advantages BBs: higher total discounted revenue, stops exploration if it has no added value

Answers to polls