Multi-Armed Bandits

Principles of Online Decision-Making (CS-303)

Prof. Matthias Grossglauser

Information and Network Dynamics (INDY) lab
School of Computer and Communication Sciences (I&C)
EPFL

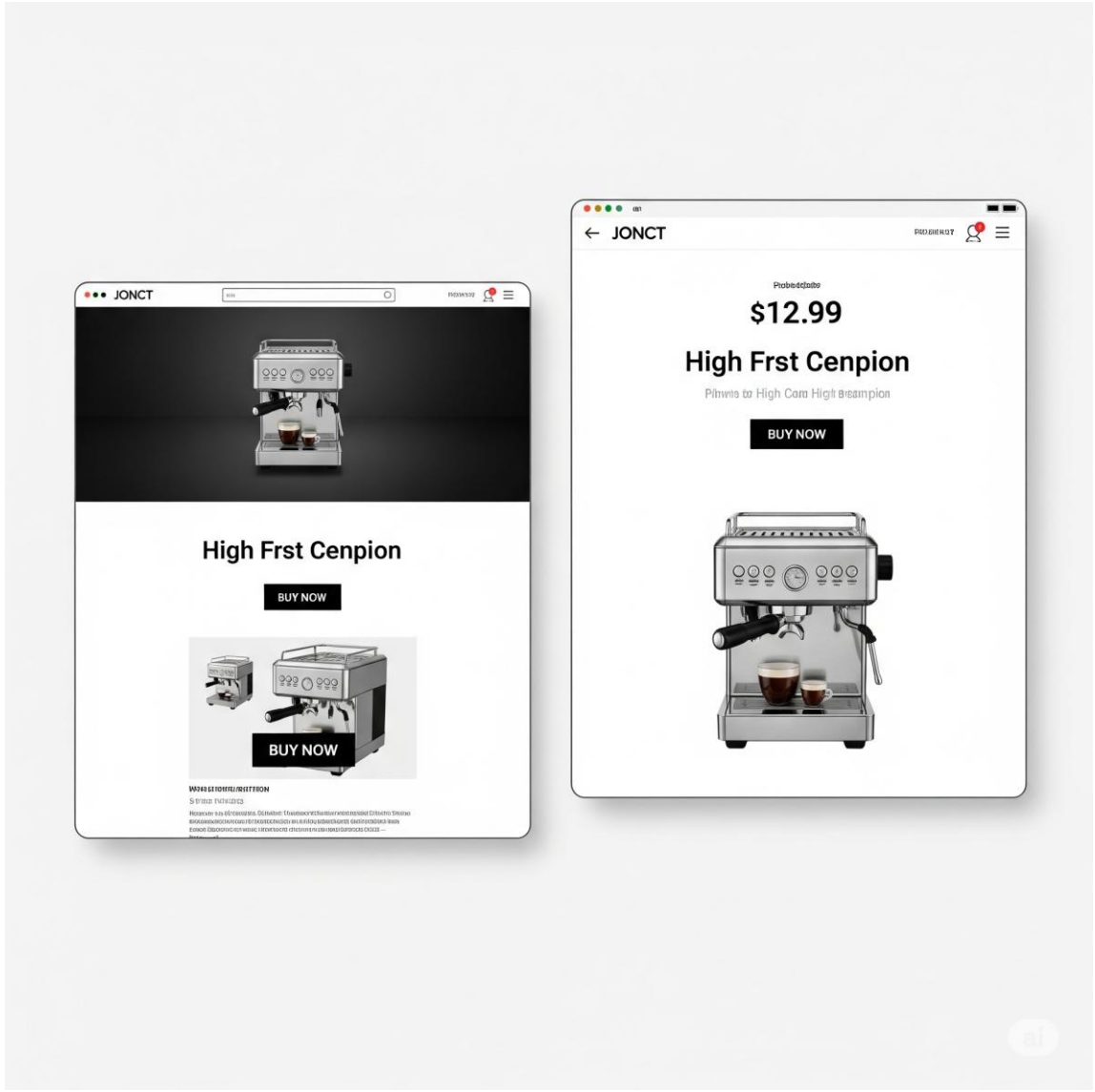
Which arm to pull?

- Slot machine:
  - Pull arm, get a reward (- the investment)
- Why bandit?
  - $\mathbb{E}[\text{payoff}] < 0$!
- In the old days: a bandit used a mechanical random number generator
  - Subtle differences between one-armed bandits
- Which bandit (arm) to play?
- In particular, if we play for a long time but know nothing at the beginning → how to use information optimally?

Multi-armed bandits: a bit of history

- William R Thompson [Biometrika, 1933]:
  - Clinical trials: minimize unnecessary harm
- Sequential decision-making under uncertainty
- Many applications in online services:
  - Lots of data, lots of decisions, human intervention in general not feasible
- Objective: principles → clean vignettes of (often messier) real-world situations
  - Mathematical tractability
  - Algorithms not necessarily directly applicable to all real-world situations…
  - … but the principles are tools guiding towards real-world solutions

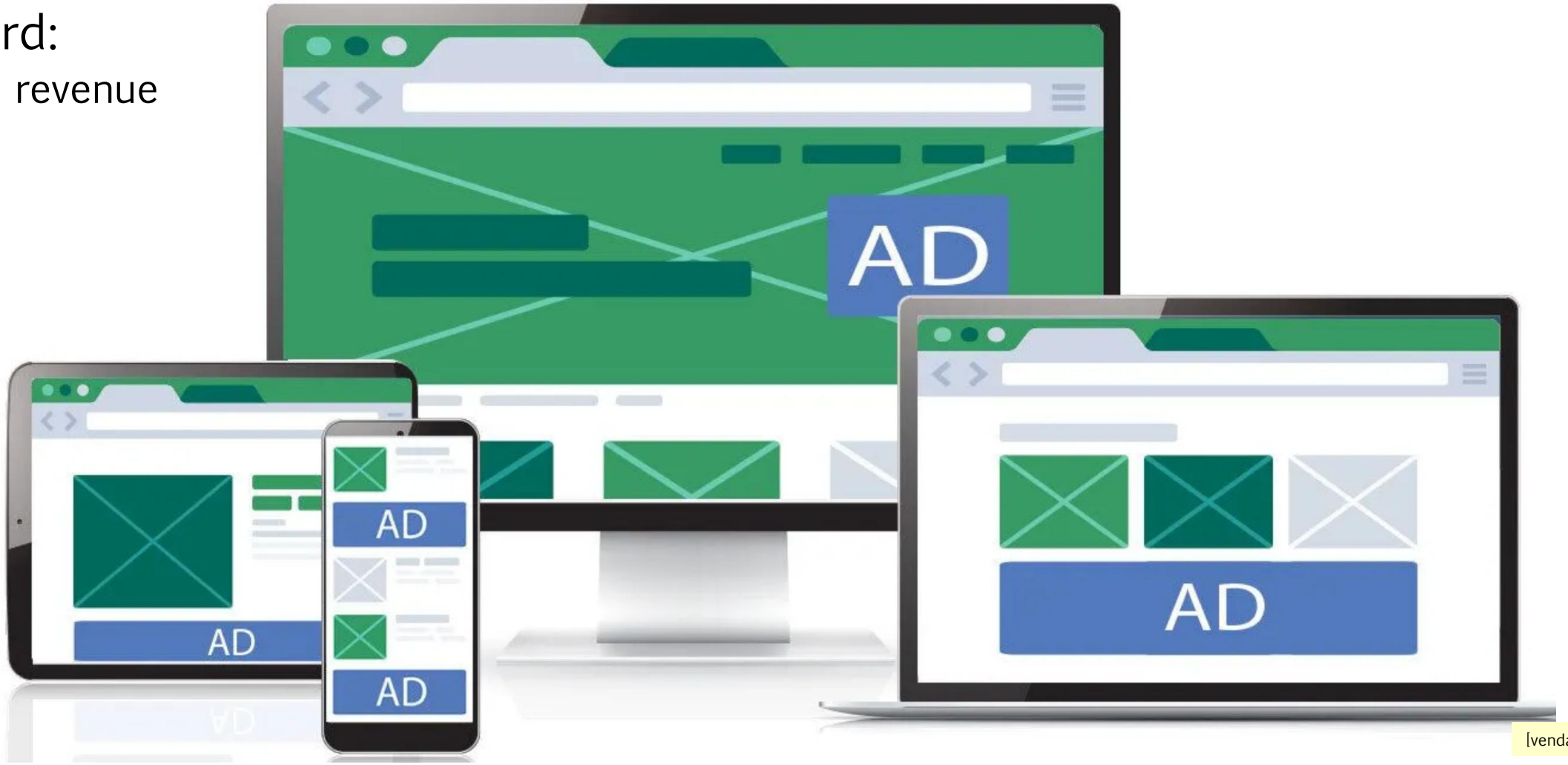
Real-world example #1: A/B-testing web interfaces



- Goal:
  - Automate making design choices for web/online interfaces
- Reward:
  - User traffic, user activity, engagement time,...

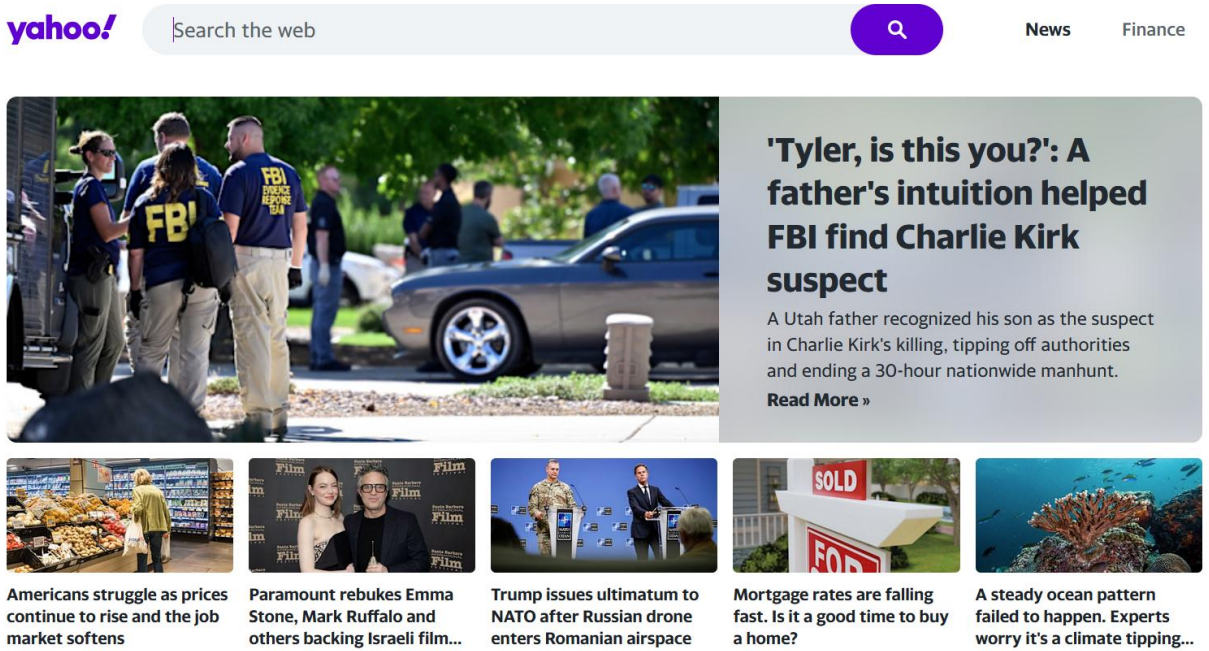
Real-world example #2: ad placement

- Goal:
  - Place the right ads at the right place, and show the right ad
- Reward:
  - Ad revenue



[vendasta.com]

Real-world example #3: news recommendations



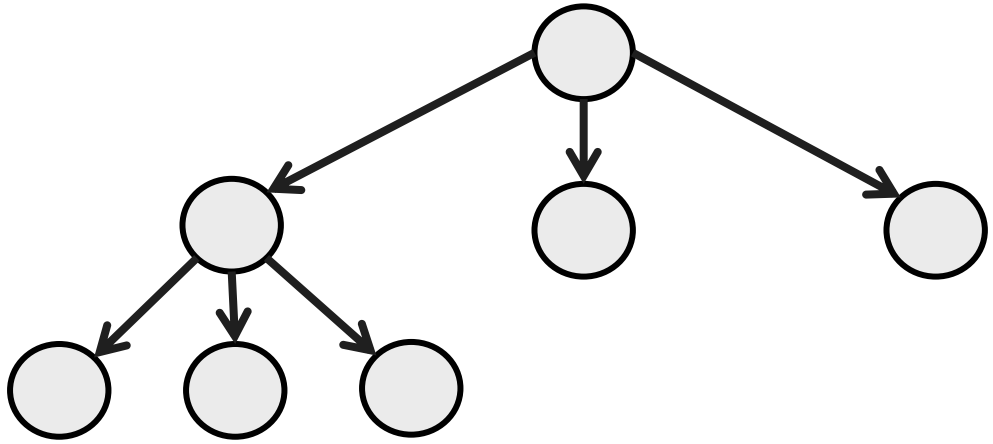
- Goal:
  - Serving up news stories of interest to every user
- Reward:
  - User click; engagement time;...

Real-world example #4: dynamic pricing

- Goal:
  - Optimize price over time so as to sell full inventory and max revenue
- Reward:
  - Total revenue

Real-world example #5: Monte Carlo tree search

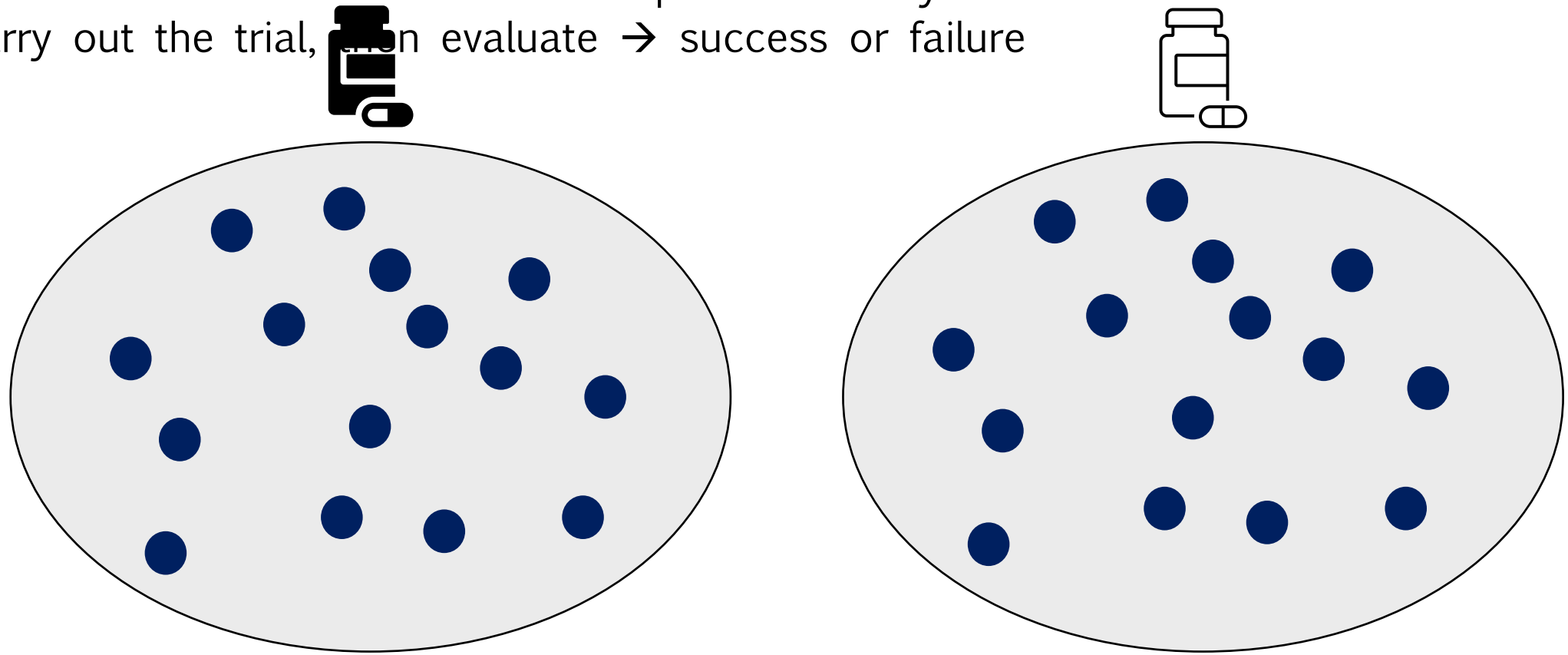
- Recent success in ML approaches to game playing (Atari games, chess, Go,…)
- Key ingredient: Monte Carlo tree search
- Tree: captures positions/situations already explored by algorithm



- Algorithm: generate a path down the known tree; expand leaf; play randomly to the end; propagate back reward
- Conceptually: every node contains a bandit algorithm: learns rewards from "further down" based on choice of child
  - Over time, more high-rewards paths get chosen, but must handle uncertainty

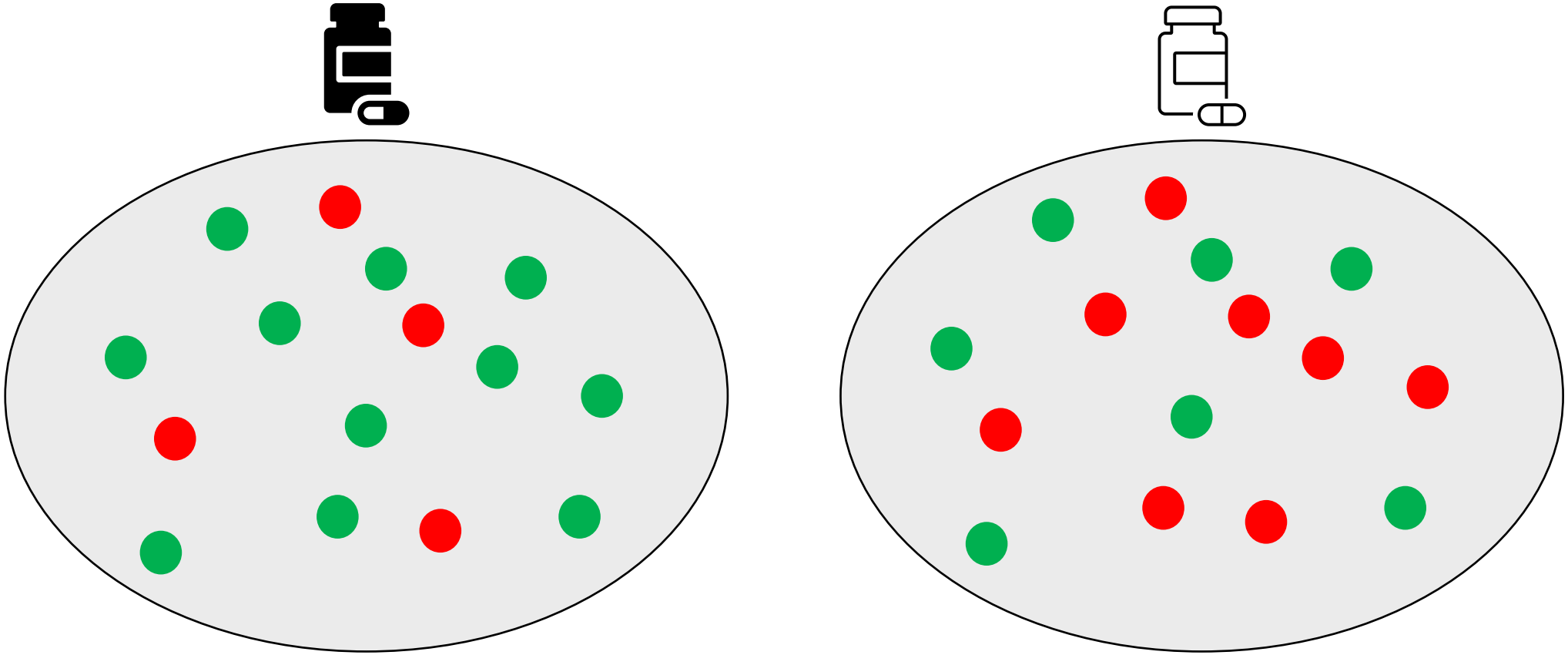
Back to clinical trial: what might go wrong?

- Randomized test: each patient randomly gets the drug or the placebo
- At the end of the trial, we assess effectiveness of treatment:
  - For example, hypothesis test: fix a p-value (e.g. 0.05) and criterion (e.g., probability of recovery twice as high as without treatment)
  - This determines the number of samples necessary
  - Carry out the trial, then evaluate → success or failure



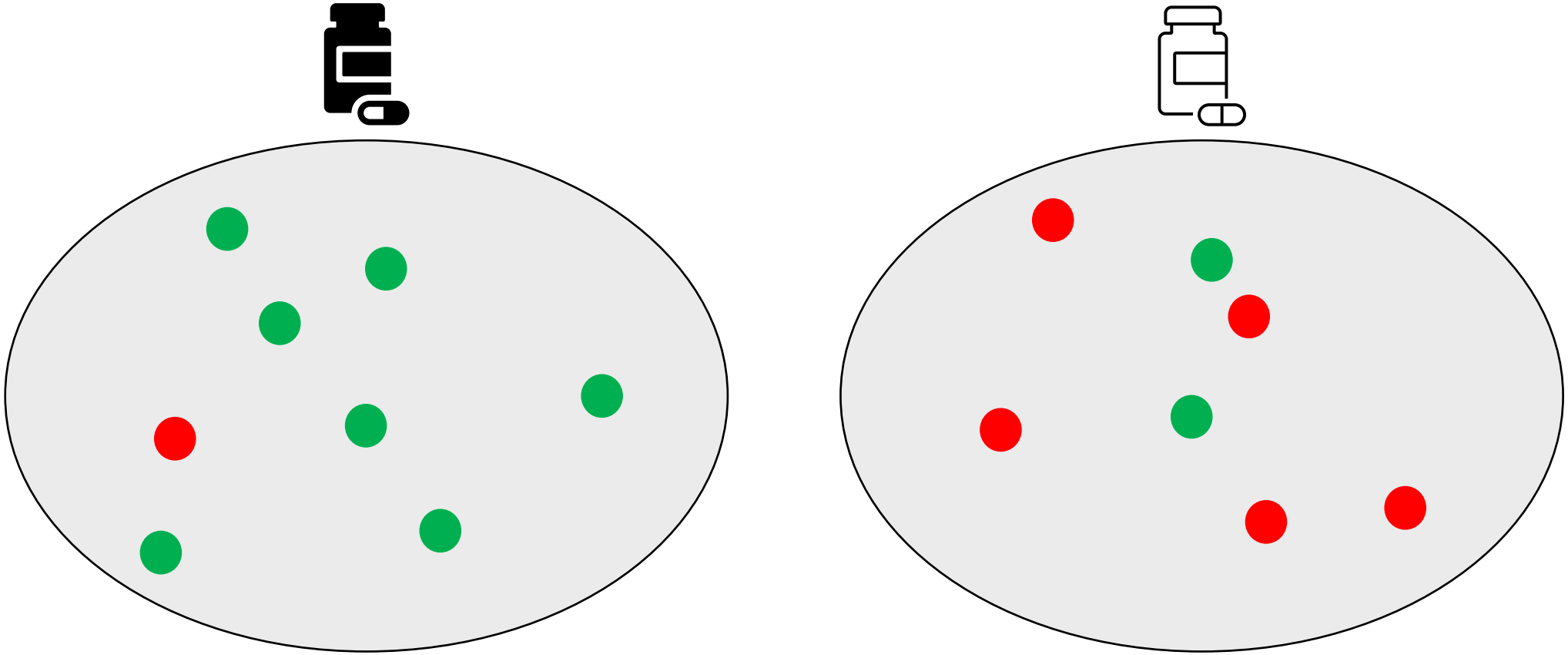
Example: clinical trial

- Randomized test: each patient randomly gets the drug or the placebo



Example: clinical trial

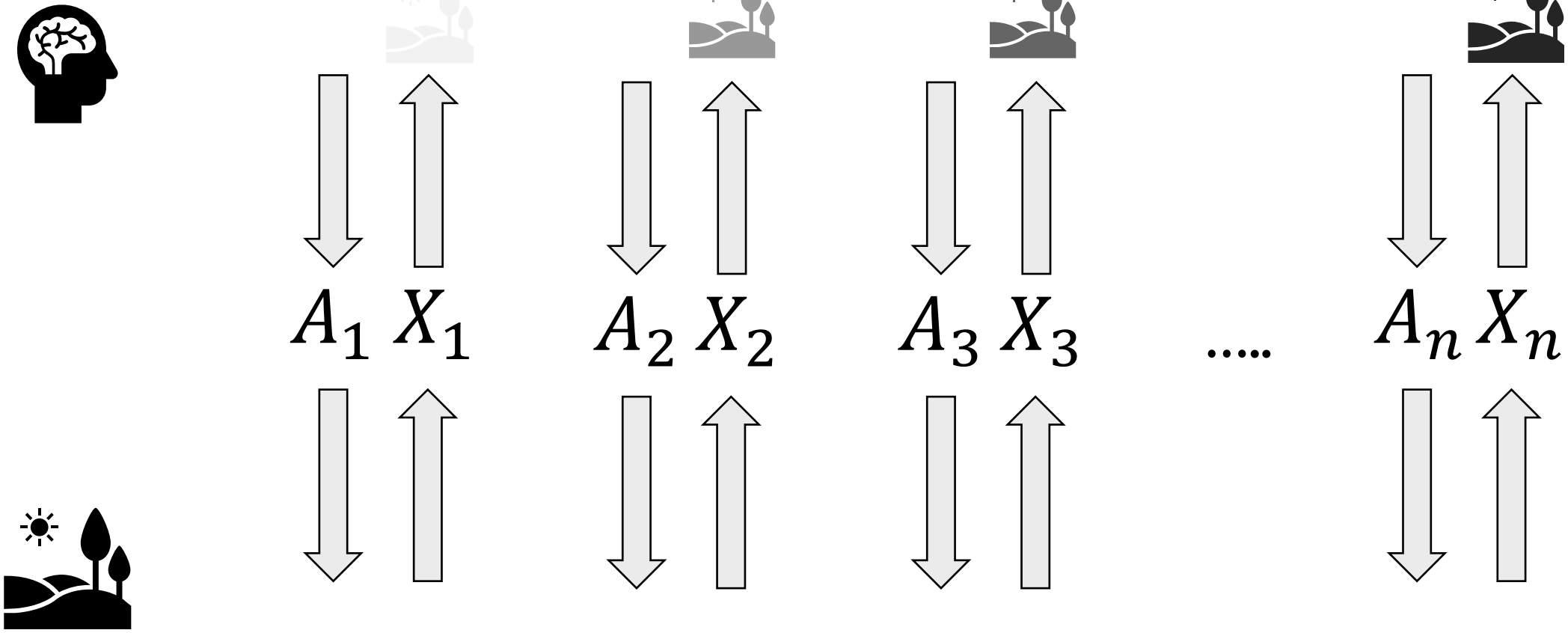
- What if after a short time, this is the situation?



Stochastic bandit: definitions, assumptions, vocabulary

Learner: takes actions (A_i)

horizon n



Environment: generates rewards (X_i)

Stochastic bandits: definitions, assumptions, vocabulary

- History $H_t = (\mathrm{A}_1, \mathrm{X}_1, \mathrm{A}_2, \mathrm{X}_2, \ldots, \mathrm{A}_\mathrm{t}, \mathrm{X}_\mathrm{t})$
  - It contains all the information the learner has so far about the environment
- Policy π:
  - Map from history to action: $A_t = \pi(H_{t-1})$
  - More specifically, action can be random → π_t is a conditional distribution over A_t given $A_1, X_1, A_2, X_2, \ldots, A_{t-1}, X_{t-1}$ (but cannot depend on the future)
- Environment:
  - Map from $(A_1, X_1, A_2, X_2, \ldots, A_t) \to X_t$
- Goal: maximize cumulative reward up to horizon n
  - $S_n = \sum_{i=1}^{n} X_i$
- Key assumption: learner does not know the environment at the beginning
  - But might know the class of possible environments
- Regret of learner relative to a policy π^*
  - Expected reward under π^* - expected reward for learner
  - "Cost of ignorance"
  - Usually: π^* = optimal algorithm for environment (i.e., knows environment perfectly)

Unstructured bandits

- Intuitively, an unstructured bandit is one where pulling one arm provides no information on the reward distribution of other arms
- Environment class: $\mathcal{E} = \{\nu = (P_a : a \in \mathcal{A}) : P_a \in \mathcal{M}_a \text{ for all } a \in \mathcal{A}\}$
  - $\mathcal{M}_a$ is a family (set) of possible distributions P_a for action a
  - In other words, $\mathcal{E} = \times_{a \in \mathcal{A}} \mathcal{M}_a$: product space
    - I.e., any combination of elements from $\mathcal{M}_a$'s is possible
  - This ensures that knowing anything about P_a provides no information on $P_{a'}$

Examples of unstructured bandit classes

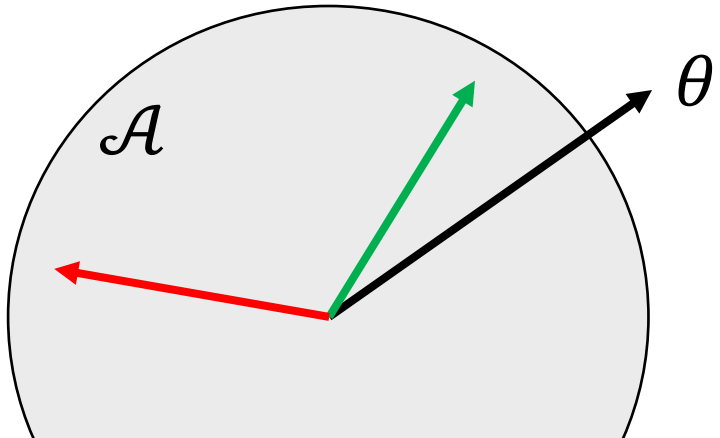
Name	Parametric	Definition
Bernoulli	Yes	$\mathcal{B}(\mu_i), \mu_i \in [0,1]$
Uniform	Yes	$\mathcal{U}(a_i, b_i), a_i \leq b_i \in \mathbb{R}$
Gaussian, known variance	Yes	$\mathcal{N}(\mu_i, \sigma^2)$
Gaussian, unknown variance	Yes	$\mathcal{N}(\mu_i, \sigma_i^2)$
Finite variance	No	$\mathbb{V}[P_i] \leq \sigma^2$
Bounded support	No	$\mathrm{Supp}(P_i) \subseteq [a, b]$
Subgaussian	No	P_i is σ-subgaussian

Structured bandits

- Intuitively, a structured bandit has an environment class where the reward of one arm provides information on reward distribution of other arms
- Examples:
  - "Complementary Bernoulli" bandit: $\mathcal{A} = \{1,2\},\ \mathcal{E} = \{(\mathcal{B}(\theta), \mathcal{B}(1-\theta) : \theta \in [0,1]\}$
    - Note: could learn by playing only one arm forever
    - Example: two mutually exclusive options to click on a webpage
  - Stochastic linear bandit: $\mathcal{A} \subset \mathbb{R}^d$ (note: uncountably infinite action space), $\theta \in \mathbb{R}^d$ $(\mathcal{E} = \{\nu_\theta : \theta \in \mathbb{R}^d\})$

$$\nu_\theta = (\mathcal{N}(< a, \theta >, 1) : a \in \mathcal{A})$$

    - Example: recommender system, θ captures unknown user preferences, a captures feature vector of items



Thought experiment: case of no noise

- Unstructured bandit:
  - Try each of k arms once, observe constant reward
  - Then play best arm forever → no more regret
  - Constant cumulative regret
- Structured bandit:
  - Bernoulli: one permanent winner, one permanent loser → pull once, then perfect forever
  - Stochastic linear: getting exact inner products, i.e., projections → set of actions needs to span $\mathbb{R}^d$ → d pulls are enough to estimate θ perfectly
- This is not very interesting, and not realistic in many practical settings
  - The main topic in bandits is in dealing with noise in the reward

Review of some probability notions

- Cumulative distribution function of a continuous RV: $F_A(a) = \mathbb{P}(A \leq a)$
  - Complementary cumulative distribution function (CCDF) = $F'_A(a) = \mathbb{P}(A > a) = 1 - F_A(a)$
  - Density: $f_A(a) = \frac{dF_A(a)}{da} \geq 0$
- Independence:
  - Two random variables are independent iff $F_{A,B}(a,b) = F_A(a)F_B(b)$
    - Information on one variable provides no information on the other variable
  - A set of RVs are pairwise independent iff every pair is independent
    - Information on one variable provides no information on a second variable, assuming we do not know anything about the other variables
  - A set of RVs are mutually independent iff $F_{A,B,C}(a,b,c) = F_A(a)F_B(b)F_C(c)$
    - Any information on all the other variables provides no information on a target variable(s)

Review of some probability notions

- Mutual is stronger than pairwise independence!
  - Example: $A, B \sim \mathcal{B}\left(\frac{1}{2}\right)$ (i.i.d.), $C = A \otimes B$ ("xor"): all pairs independent, but not mutually independent!
    Contrast with $A, B, C \sim \mathcal{B}\left(\frac{1}{2}\right)$ (i.i.d.)

A	B	ℙ
0	0	1/4
0	1	1/4
1	0	1/4
1	1	1/4

B	C	ℙ
0	0	1/4
0	1	1/4
1	0	1/4
1	1	1/4

C	A	ℙ
0	0	1/4
0	1	1/4
1	0	1/4
1	1	1/4

A	B	C	ℙ
0	0	0	1/4
0	0	1	0
0	1	0	0
0	1	1	1/4
1	0	0	0
1	0	1	1/4
1	1	0	1/4
1	1	1	0

A	B	C	ℙ
0	0	0	1/8
0	0	1	1/8
0	1	0	1/8
0	1	1	1/8
1	0	0	1/8
1	0	1	1/8
1	1	0	1/8
1	1	1	1/8

Review of some probability notions

- Conditional independence:
  - Discrete: $\mathbb{P}(A \leq a, B \leq b | C = c) = \mathbb{P}(A \leq a | C = c)\mathbb{P}(B \leq b | C = c)$ for all a, b, c (where $\mathbb{P}(C = c) > 0$)
  - Continuous: $f_{AB|C}(a, b|c) = f_{A|C}(a|c) f_{B|C}(b|c)$ for all a, b, c (where $f_C(c) > 0$)
  - Intuitively: if C is known, then A provides no further information about B (and vice versa)

  - Notions of pairwise and mutual independence also apply to conditional independence, when there are more than two variables

Review of some probability notions

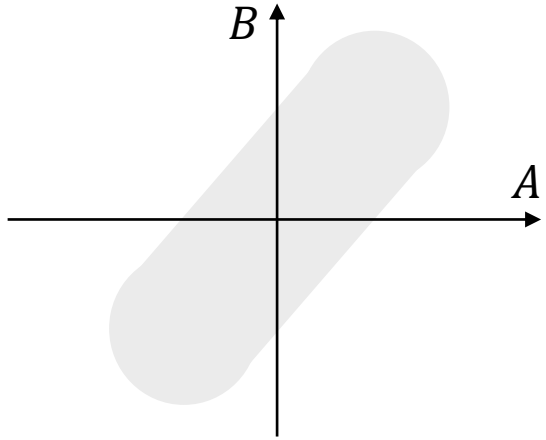
- Expectation (or expected value, or mean):
  - $\mathbb{E}[X] = \mu_X = \begin{cases} \sum_{x_i} x_i \mathbb{P}(X = x_i) & \text{discrete} \\ \int x f_X(x) dx & \text{continuous} \end{cases}$
  - Some important properties:
    - Sum: $\mathbb{E}[A + B] = \mathbb{E}[A] + \mathbb{E}[B]$ for any A, B (linearity of expectation)
    - Product: $\mathbb{E}[AB] = \mathbb{E}[A]\mathbb{E}[B]$ if $A \perp B$, but not true in general
  - For $X \geq 0$: $\mathbb{E}[X] = \int_0^\infty F'(x) dx$ (ccdf)
  - $\mathbb{E}[\mathbb{I}\{E\}] = \mathbb{P}(E)$, where $\mathbb{I}(E)$ is the indicator for event E

Review of some probability notions

- Conditional expectation:
  - $\mathbb{E}[A|B=b] = \sum_a a\mathbb{P}(A=a|B=b) \stackrel{\text{def}}{=} f(b)$
  - $\mathbb{E}[A|B] = f(B)$
    - Note: this is a random variable in B!
  - Example:
    - $B \sim \text{unif}(0,1)$
    - $A \sim \mathcal{B}(B)$
    - Then $\mathbb{E}[A|B] = B$

  - If $A \perp B$, then $\mathbb{E}[A|B] = \mathbb{E}[A]$
  - $\mathbb{E}[Ag(B)|B] = g(B)\mathbb{E}[A|B]$: conditional on $B=b$, $g(B)=g(b)$ is a constant and can be "pulled in front" of the sum/integral
  - Total expectation: $\mathbb{E}\big[\mathbb{E}[A|B]\big] = \mathbb{E}[A]$

Law of total expectation (tower property)



- Example:
  - Suppose we have two random variables defined as follows:
    - $C \sim \text{unif}(-1,1)$
    - $A = \mathcal{N}(C, 1)$
    - $B = \mathcal{N}(C, 1)$
    - and $A \perp B|C$
      - Or: $A - C - B$ is a Markov chain
  - Note that A, B are dependent (even though conditionally independent)
  - Suppose we need to compute $\mathbb{E}[AB]$ - what's an easy way to do this?
    - Could find the joint density $f_{AB}(a, b)$ of A and B, then compute $\iint ab\, f_{AB}(a, b)\, da\, db$
    - Or we can use the law of total expectation (tower property):
      - $\mathbb{E}[AB] = \mathbb{E}[\mathbb{E}[AB|C]]$
      - Note that because of conditional independence,
        $$\mathbb{E}[AB|C = c] = \mathbb{E}[A|C = c]\mathbb{E}[B|C = c] = c^2$$
      - $\mathbb{E}\big[\mathbb{E}[AB|C]\big] = \mathbb{E}[C^2] = \int_{-1}^{1} \frac{1}{2} c^2 dc = \frac{1}{3}$

Back to bandits: regret

- Expected reward and maximum reward:
  - $\mu_a(v) = \int_{-\infty}^{+\infty} x \, dP_a(x) = \mathbb{E}[X]$ with $X \sim P_a$
  - $\mu^*(v) = \max_{a \in \mathcal{A}} \mu_a(v)$

- Stochastic bandit given by $\nu = (P_a : a \in \mathcal{A})$
- Regret of policy π on bandit instance ν: $R_n(\pi, \nu) = n\mu^* - \mathbb{E}[\sum_{t=1}^{n} X_t]$

- Lemma:
  - (a) $R_n(\pi, \nu) \geq 0$ for all policies π
  - (b) The policy π choosing $A_t \in \arg\max_a \mu_a$ for all t satisfies $R_n(\pi, \nu) = 0$
  - (c) If $R_n(\pi, \nu) = 0$ for some policy π, then $\mathbb{P}(\mu_{A_t} = \mu^*) = 1$ for all $t \in [n]$

Objective of the learner

- Linear regret $\theta(n)$ is trivial: no need to learn anything
- Slightly more ambitious objective: sublinear regret
  For all $\nu \in \mathcal{E}$, $\lim_{n\to\infty} \frac{R_n(\pi,\nu)}{n} = 0$ (also written $R_n = o(n)$)
  - This means that in the long run, the learner chooses the optimal action almost always
- Usually we are interested in more ambitious objectives
  - In general, learner needs to discover the/an arm with largest mean
- For this, it needs to pull **each** arm (unstructured) or **some** number of arms (structured) a number of times to estimate environment parameters
  - Tradeoff between exploring (playing all arms to improve estimators) vs exploiting (playing the best arm(s))

Decomposing the regret

- Suboptimality gap of action a:
  - $\Delta_a(\nu) = \mu^*(\nu) - \mu_a(\nu)$
- Action count of action a by time t:
  - $T_a(t) = \sum_{s=1}^{t} \mathbb{I}\{A_s = a\}$
    - Note: $T_a(t)$ is a random variable, because it depends on the random rewards observed before t
- Lemma: regret decomposition:
  for any policy π and environment ν, the regret satisfies

$$R_n(\pi, \nu) = \sum_{a \in \mathcal{A}} \Delta_a \, \mathbb{E}[T_a(n)]$$

Proof of regret decomposition lemma

- Rewriting the n-step regret:

$$R_n = n\mu^* - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right] = \sum_{a\in\mathcal{A}}\sum_{t=1}^{n} \mathbb{E}\left[(\mu^* - X_t)\mathbb{I}(A_t = a)\right]$$

$$= \sum_{a\in\mathcal{A}}\sum_{t=1}^{n} \mathbb{E}\Big[\mathbb{E}[(\mu^* - X_t)\mathbb{I}(A_t = a)|A_t]\Big]$$

- Note: we use $\mathbb{E}[AB] = \mathbb{E}\Big[\mathbb{E}[AB|C]\Big]$

Proof of regret decomposition lemma

$$\mathbb{E}[(\mu^* - X_t)\mathbb{I}(A_t = a)|A_t] = \mathbb{I}(A_t = a)\mathbb{E}[\mu^* - X_t|A_t]$$

Conditional on A_t, $\mathbb{I}(A_t = a)$ is constant

$$= \mathbb{I}(A_t = a)(\mu^* - \mu_{A_t})$$

By assumption, $\mathbb{E}[X_t|A_t] = \mu_{A_t}$

$$= \mathbb{I}(A_t = a)(\mu^* - \mu_a)$$

Given that first factor is nonzero only for $A_t = a$, we can replace μ_{A_t} with μ_a

$$= \mathbb{I}(A_t = a)\Delta_a$$

Proof of regret decomposition lemma

- Rewriting the n-step regret:

$$R_n = n\mu^* - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right] = \sum_{a\in\mathcal{A}}\sum_{t=1}^{n} \mathbb{E}[\mathbb{I}(A_t = a)\Delta_a] = \sum_{a\in\mathcal{A}} \Delta_a\, \mathbb{E}[T_a(n)]$$

- Intuition:
  - The learner should pull arms with low or zero suboptimality gap, and minimize expected number of pulls of arms with high suboptimality gap
  - This is not exactly shocking ;) The game will be in devising algorithms that get there efficiently

Summary

- What have we learned:
  - Notion of the stochastic bandit: player and environment interact via actions and rewards
  - The rewards are instantaneous
  - Different assumptions about the possible environments are possible – this assumption is very important, because the player needs to figure out the rest by probing (i.e., exploring)
  - Regret: how much do we lose (over the horizon n) because we do not know the environment?
  - Decomposition of the regret into a sum over all arms of their suboptimality gaps times number of times played
- Reading assignment:
  - L&S: 4.1, 4.2, 4.3, 4.4, 4.5