Lecture 12: Contextual and Structured Online Learning

- Structured bandits
- Contextual bandits
- OFUL/Kernel-UCB/Kernel-TS

Recall: Stochastic bandit setting

- Set $\mathcal{K} = \{1, 2, \ldots, K\}$ of K actions (arms, machines)
- You are facing a tuple of distributions $\nu = (P_1, P_2, \ldots, P_K)$

P_1 $\quad$ P_2 $\quad \ldots \quad$ P_K

- Identify the best action by interacting with the environment

Recall: Stochastic bandit game

- Set $\mathcal{K} = \{1, 2, \ldots, K\}$ of K actions (arms, machines)
- You are facing a tuple of distributions $\nu = (P_1, P_2, \ldots, P_K)$
- Distribution P_k under tuple ν has expectation $\mu_k(\nu)$
- For each round t:
  1. Select an action $k_t \in \mathcal{K}$
  2. Play action k_t
  3. Observe reward $r_t \sim P_{k_t}$

Goal: Maximize $\sum_{t=1}^{T} \mu_{k_t}(\nu) \rightarrow$ play $k_\star = \arg\max_{k \in \mathcal{K}} \mu_k(\nu)$

Recall: Regret

Minimize regret:

$$R_T(\pi, \nu) = T\mu_\star(\nu) - \sum_{t=1}^{T} \mu_{k_t}(\nu) \qquad \text{where } \mu_\star(\nu) = \max_{k \in \mathcal{K}} \mu_k(\nu)$$

Decomposing the regret:

- Suboptimality gap: $\Delta_k(\nu) = \mu_\star(\nu) - \mu_k(\nu)$
- Number of plays of action k up to time t: $N_k(t) = \sum_{s=1}^{t} \mathbb{I}\{k_s = k\}$

$$R_T(\pi, \nu) = \sum_{k \in \mathcal{K}} \Delta_k(\nu)\, \mathbb{E}[N_k(T)]$$

What happens when K is very large?

Exploit structure

- Learn about an action *without trying it*!
- Requires a notion of similarity between actions
- Example:
  - Online parameter tuning (velocities, powers, intensities)
  - Treatment optimization (volumes, quantities, time)
  - Recommender systems

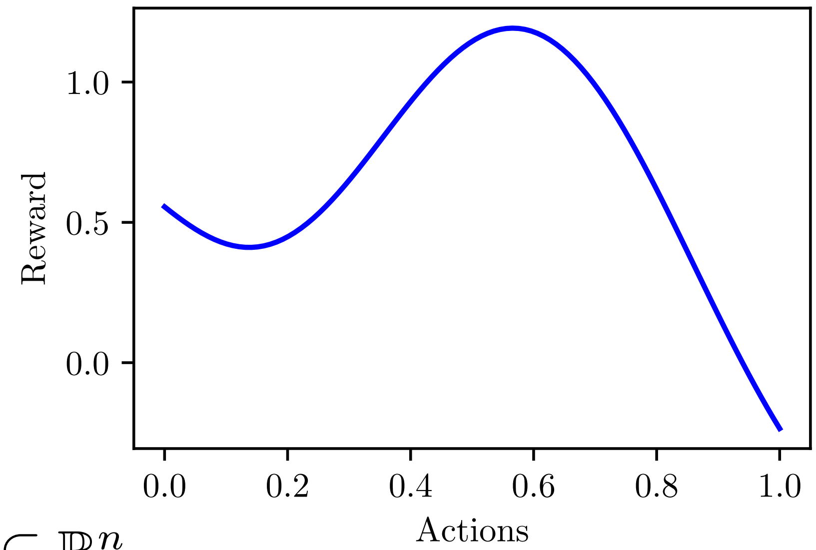
Example: Photo recommendation

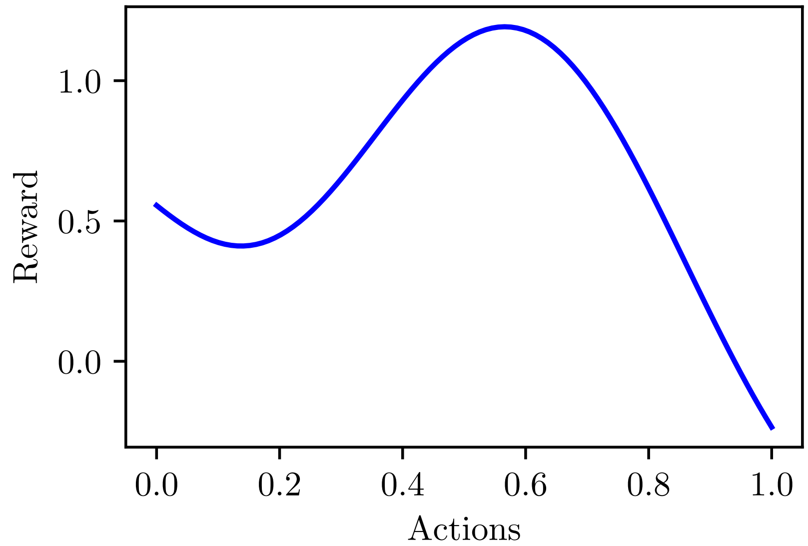
Stochastic bandit with structured actions



- Action space $\mathcal{X} \subseteq \mathbb{R}^n$
- Reward function $f : \mathcal{X} \mapsto \mathbb{R}$
- For each round t:
  1. Select an action $x_t \in \mathcal{X}$
  2. Play action x_t
  3. Observe reward $r_t = f(x_t) + \epsilon_t$ $\quad \leftarrow$ Observation noise ϵ_t

Goal: Maximize $\sum_{t=1}^{T} f(x_t) \rightarrow$ play $x_\star = \arg\max_{x \in \mathcal{X}} f(x)$

Online function approximation



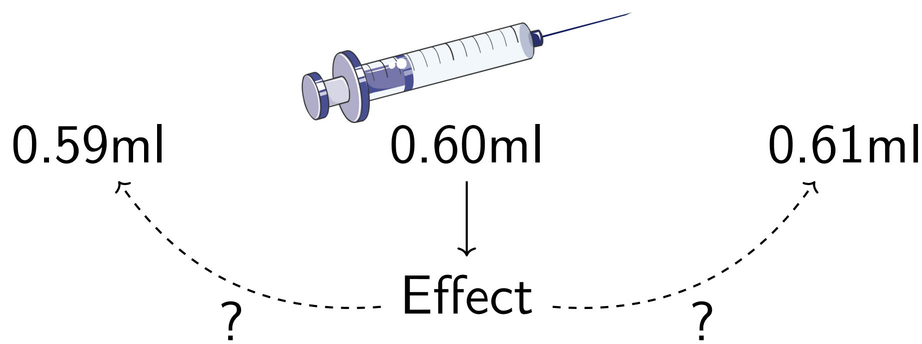
- Sequentially select locations where to observe the function
- Noisy observations
- Gathering an observation is not *free*

Example: Adaptive treatment dosage

What is the best treatment dosage for some disease?

- Space $\mathcal{X}$ of possible dosages
- Patients come in sequentially
- For each patient $t \geq 1$:
  1. Select a dosage $x_t \in \mathcal{X}$
  2. Treat patient t with dosage x_t
  3. Observe the effectiveness $r_t = f(x_t) + \epsilon_t$

Goal: Maximize the effectiveness at every step $\sum_{t=1}^{T} f(x_t)$



The linear case

- Action space $\mathcal{X} \subseteq \mathbb{R}^n$
- There exists an **unknown** parameter $\theta_\star \in \mathbb{R}^n$ such that $f(x) = \langle x, \theta_\star \rangle$
- On each rount t:
  1. Select an action $x_t \in \mathcal{X}$
  2. Play action x_t
  3. Observe reward $r_t = \langle x_t, \theta_\star \rangle + \epsilon_t$
- Goal: maximize $\sum_{t=1}^{T} f(x_t) \rightarrow$ play $x_\star = \arg\max_{x \in \mathcal{X}} \langle x, \theta_\star \rangle$

$\rightarrow$ Minimize

$$R_T(\pi, \theta_\star) = \sum_{t=1}^{T} \langle x_\star, \theta_\star \rangle - \sum_{t=1}^{T} \langle x_t, \theta_\star \rangle = \sum_{t=1}^{T} \langle x_\star - x_t, \theta_\star \rangle$$

Typical assumptions

- Action space $\mathcal{X}$ lies in a bounded set
  What would happen if it did not?
- Noise ϵ_t satisfies $\mathbb{E}[\epsilon_t | x_{1:t}, \epsilon_{1:t}] = 0$ and tail-constraints
- More specifically, ϵ_t is R-subGaussian for a fixed constant $R \geq 0$
  – A real-valued random variable X is R-subgaussian if

$$\mathbb{E}\left[e^{\gamma X}\right] \leq e^{\gamma^2 R^2 / 2}$$

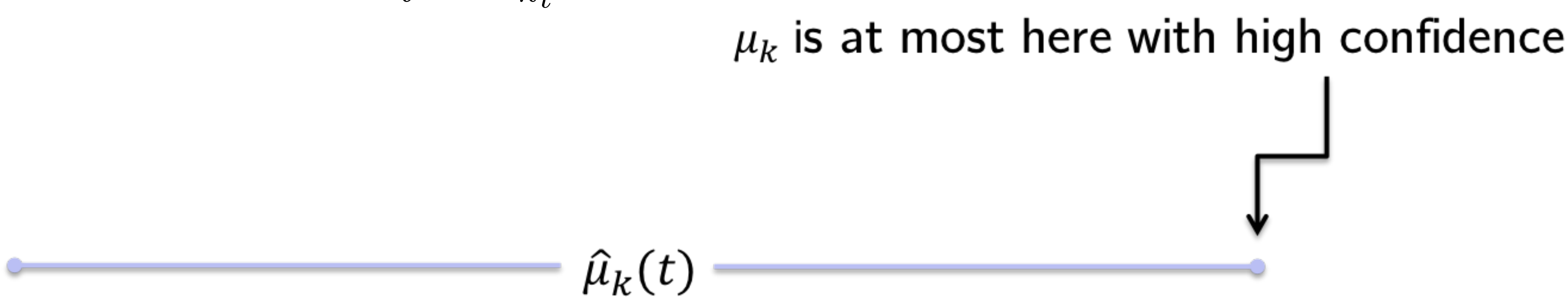
$\rightarrow$ The Laplace transform of X is dominated by the Laplace transform of a random variable sampled from $\mathcal{N}(0, R^2)$

  – Requires that the tails of the noise distribution are dominated by the tails of a Gaussian distribution
  – For example, true for: Gaussian noise, bounded noise

Recall: UCB algorithm

$$\mathrm{UCB}_k(t,\delta) = \hat{\mu}_k(t) + \sqrt{\frac{2\ln(1/\delta)}{N_k(t)}}$$

- Action set $\mathcal{K} = \{1, 2, \ldots, K\}$, confidence level δ
- Play each action once
- For each round $t > K$:
  1. Select action $k_t = \arg\max_{k \in \mathcal{K}} \mathrm{UCB}_k(t-1, \delta)$
  2. Play action k_t
  3. Receive reward $r_t \sim P_{k_t}$



Optimism in the Face of Uncertainty principle (OFU)

- Maintain a confidence set $C_{t-1} \subseteq \mathbb{R}^n$ for the parameter $\theta_\star$
- Calculate C_{t-1} from $x_1, r_1, x_2, r_2, \ldots, x_{t-1}, r_{t-1}$ such that $\theta_\star \in C_{t-1}$ with high probability

  Confidence sets generalize confidence intervals to multiple dimensions

- Each parameter θ in C_{t-1} is *potentially* $\theta_\star$
- For each θ in C_{t-1}: if this θ is $\theta_\star$, what would be $f(x_{\star,\theta})$?

  $\rightarrow$ $x_{\star,\theta} = \arg\max_{x\in\mathcal{X}} \langle x, \theta \rangle$
  $\rightarrow$ $f(x_{\star,\theta}) = \max_{x\in\mathcal{X}} \langle x, \theta \rangle$
- Optimistic $\tilde{\theta}_t = \arg\max_{\theta \in C_{t-1}} f(x_{\star,\theta})$

OFUL algorithm

- **OFU** for **L**inear bandits
- Action space $\mathcal{X} \subseteq \mathbb{R}^n$
- Reward function $f(x) = \langle x, \theta_\star \rangle$
- On each round t:
  1. Choose an optimistic estimate $\tilde{\theta}_t = \arg\max_{\theta \in C_{t-1}} (\max_{x \in \mathcal{X}} \langle x, \theta \rangle)$
  2. Select action $x_t = \arg\max_{x \in \mathcal{X}} \langle x, \tilde{\theta}_t \rangle$
  3. Play action x_t
  4. Receive reward $r_t = \langle x_t, \theta_\star \rangle + \epsilon_t$

Do you see any links between OFUL and UCB?

What if the function is non-linear?

Recall: Generalized linear regression

- Feature mapping $\phi(\cdot)$: column vector of d real numbers
- Assume $f(x) = \langle \phi(x), \theta_\star \rangle$

$\rightarrow$ $\theta_\star$ has dimension d

What if d is very large (e.g. $d \rightarrow \infty$)?

Recall: Kernel regression

- $\mathbf{y} = (r_1, r_2, \ldots, r_t)^\top$: column vector of t observations
- Feature mapping $\phi(\cdot)$; feature matrix $\mathbf{\Phi}$ of size $t \times d$

Kernel matrix $\quad \mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_t) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_t) \\ \vdots & \vdots & & \vdots \\ k(x_t, x_1) & k(x_t, x_2) & \ldots & k(x_t, x_t) \end{bmatrix}$

Kernel vector $\quad \mathbf{k}(x) = \phi(x)^\top \mathbf{\Phi}^\top = \begin{bmatrix} k(x, x_1) \\ k(x, x_2) \\ \vdots \\ k(x, x_t) \end{bmatrix}$

- The prediction at some input point x is given by

$$\hat{f}(x) = \phi(x)^\top \mathbf{\Phi}^\top (\mathbf{K} + \lambda \mathbf{I}_t)^{-1} \mathbf{y} = \mathbf{k}(x)(\mathbf{K} + \lambda \mathbf{I}_t)^{-1} \mathbf{y}$$

Recall: Bayesian view of regression

- Consider noisy observations $y = f(x) + \epsilon = \phi(x)^\top \theta_\star + \epsilon$
- With Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- With Gaussian prior on parameters $\theta_\star \sim \mathcal{N}_d(0, \Sigma_{\theta_\star})$
- The pointwise posterior predictive distribution is a normal distribution

$$\tilde{f}(x) | x_1, \ldots, x_m, y_1, \ldots, y_m \sim \mathcal{N}\left(\hat{f}(x), s^2(x)\right)$$

of expectation

$$\hat{f}(x) = \phi(x)^\top \Sigma_{\theta_\star} \mathbf{\Phi}^\top (\mathbf{\Phi} \Sigma_{\theta_\star} \mathbf{\Phi}^\top + \sigma^2 \mathbf{I}_m)^{-1} \mathbf{y}$$

and variance

$$s^2(x) = \phi(x)^\top \Sigma_{\theta_\star} \phi(x) - \phi(x)^\top \Sigma_{\theta_\star} \mathbf{\Phi}^\top (\mathbf{\Phi}^\top \Sigma_{\theta_\star} \mathbf{\Phi} + \sigma^2 \mathbf{I}_m)^{-1} \mathbf{\Phi} \Sigma_{\theta_\star} \phi(x)$$

Recall: Using prior $\Sigma_{\theta_\star} = \frac{\sigma^2}{\lambda}\mathbf{I}_d$

The predictive mean/variance rewrite as:

$$\hat{f}(x) = \phi(x)^\top \Sigma_{\theta_\star} \mathbf{\Phi}^\top (\mathbf{\Phi}\Sigma_{\theta_\star}\mathbf{\Phi}^\top + \sigma^2 \mathbf{I}_m)^{-1}\mathbf{y}$$

$$= \phi(x)^\top \frac{\sigma^2}{\lambda}\mathbf{\Phi}^\top \left(\mathbf{\Phi}\frac{\sigma^2}{\lambda}\mathbf{\Phi}^\top + \sigma^2\mathbf{I}_m\right)^{-1}\mathbf{y}$$

$$= \mathbf{k}(x)^\top (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}$$

$$s^2(x) = \phi(x)^\top \Sigma_{\theta_\star}\phi(x) - \phi(x)^\top \Sigma_{\theta_\star}\mathbf{\Phi}^\top (\mathbf{\Phi}^\top \Sigma_{\theta_\star}\mathbf{\Phi} + \sigma^2\mathbf{I}_m)^{-1}\mathbf{\Phi}\Sigma_{\theta_\star}\phi(x)$$

$$= \phi(x)^\top \frac{\sigma^2}{\lambda}\phi(x) - \phi(x)^\top\frac{\sigma^2}{\lambda}\mathbf{\Phi}^\top \left(\mathbf{\Phi}^\top \frac{\sigma^2}{\lambda}\mathbf{\Phi} + \sigma^2\mathbf{I}_m\right)^{-1}\mathbf{\Phi}\frac{\sigma^2}{\lambda}\phi(x)$$

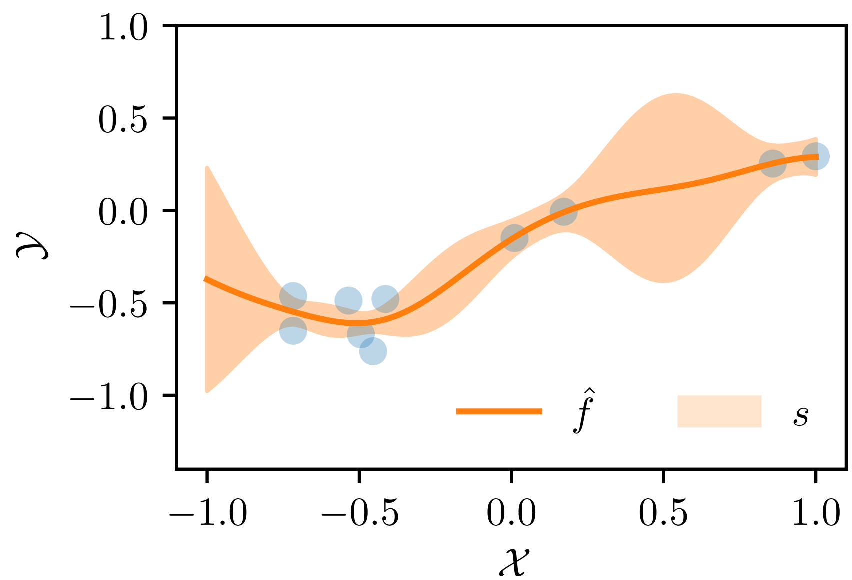
$$= \frac{\sigma^2}{\lambda}k_\lambda(x,x) \qquad \text{with}$$

$$k_\lambda(x,x') = k(x,x') - \mathbf{k}(x)^\top (\mathbf{K} + \lambda \mathbf{I}_m)^{-1}\mathbf{k}(x')$$

Recall: Gaussian Process (GP)

- By considering the covariance between *every points in the space*, we get a distribution over functions!
- Posterior distribution on f:

$$P[f|x,\mathbf{y}] \sim \mathcal{N}\left(\left[\hat{f}(x)\right]_{x\in\mathcal{X}}, \frac{\sigma^2}{\lambda}\left[k_\lambda(x,x')\right]_{x,x'\in\mathcal{X}}\right)$$



This gives predictions, but also uncertainty!

Confidence envelope

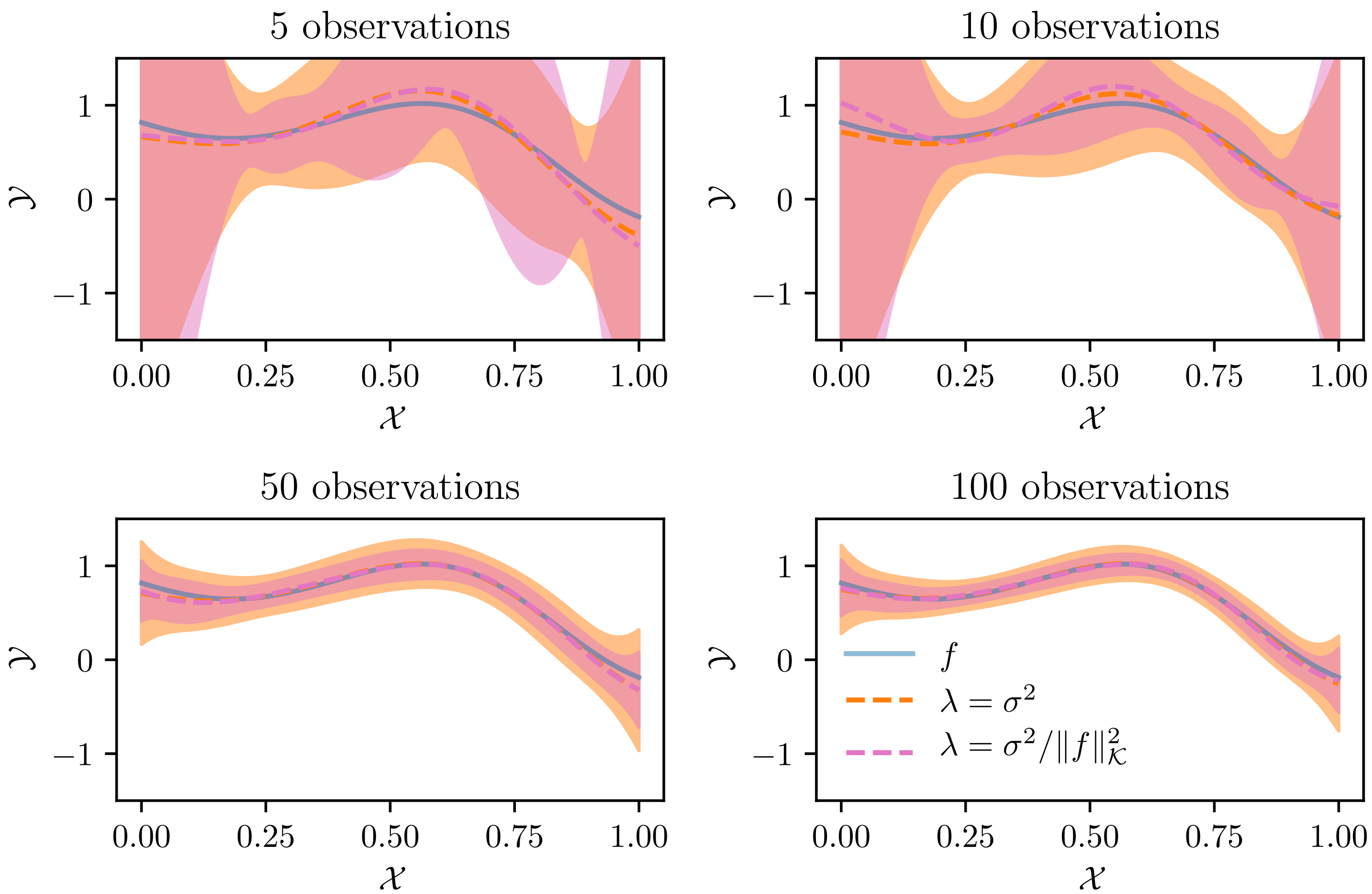
Theorem 1 (Maillard (2016)). *Assuming σ-subgaussian noise, we have*

$$|f(x) - \hat{f}_t(x)| \leq \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}}\left[\sqrt{\lambda}\|\theta_\star\|_2 + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda)}\right]$$

- *With probability higher than $1 - \delta$*
- *Simultaneously for all $t \geq 0$, for all x*
- *Recall: information gain*

$$\gamma_t(\lambda) = \sum_{s=1}^{t} \frac{1}{2} \ln\left[1 + \frac{1}{\lambda} k_{\lambda,s-1}(x_s, x_s)\right]$$

Confidence envelope



Kernel-UCB

$$\mathrm{UCB}_x(t, \lambda, \delta) = \hat{f}_t(x) + \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} \left[\sqrt{\lambda} \|\theta_\star\|_2 + \sigma \sqrt{2 \ln(1/\delta) + 2\gamma_t(\lambda)} \right]$$

- Action space $\mathcal{K} \subseteq \mathbb{R}^n$
- There exists an **unknown** parameter $\theta_\star \in \mathbb{R}^d$ such that $f(k) = \langle \phi(k), \theta_\star \rangle$
- For each round t:
  1. Select action $x_t = \arg\max_{x \in \mathcal{X}} \mathrm{UCB}_x(t, \lambda, \delta)$
  2. Play action x_t
  3. Observe reward $r_t = f(x_t) + \epsilon_t$

Act optimistically directly on $\hat{f}(x)$ rather than through $\hat{\theta}_t$

What if we wanted a stochastic approach?

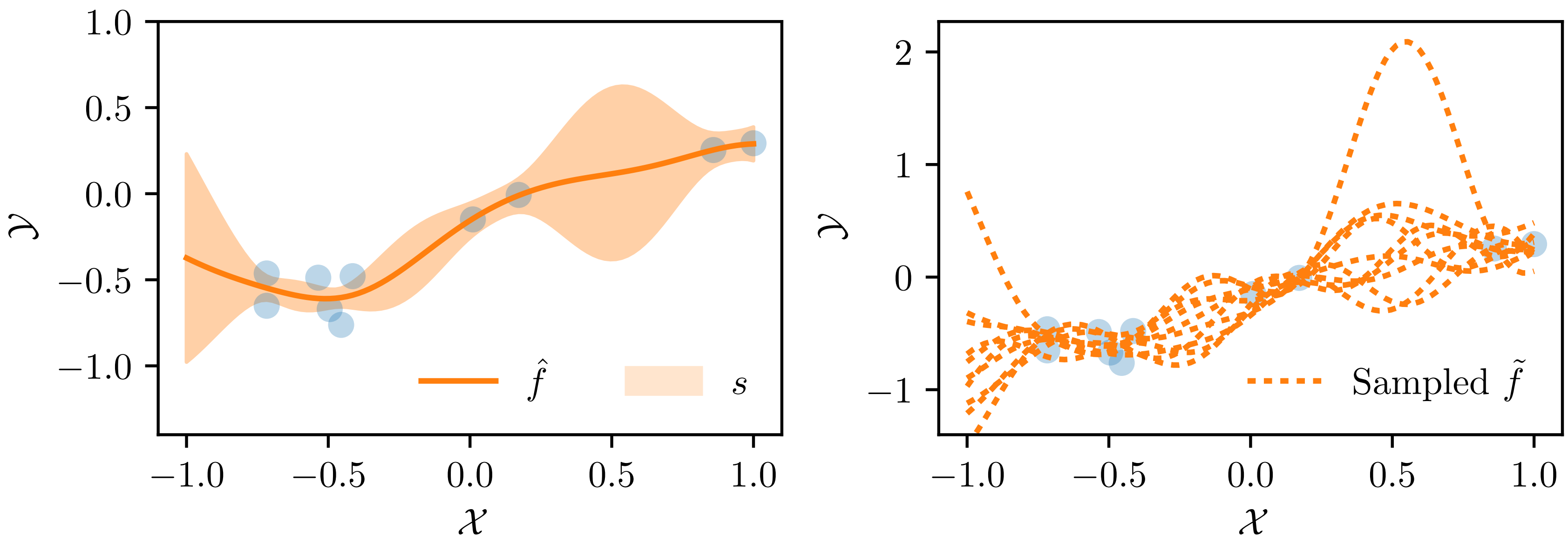
Recall: Thompson Sampling

- Action set $\mathcal{K} = \{1, 2, \ldots, K\}$
- Select next action based on its probability of being optimal
- Maintain one posterior $\pi_t^{(k)}$ for each action $k \in \mathcal{K}$
- At each round t:
  1. Sample one value $\tilde{\mu}_k \sim \pi_t^{(k)}$ for each action $k \in \mathcal{K}$
  2. Select action $k_t = \arg\max_{k \in \mathcal{K}} \tilde{\mu}_k$
  3. Play action k_t
  4. Observe reward $r_t \sim P_{k_t}$

How do we extend that to the structured setting with kernel regression?

Recall: Sampling from a Gaussian Process

- Generalization of normal probability distribution to the function space
  - From a normal distribution we sample variables
  - From a GP we sample *functions*!



Kernel-TS

- Discrete action space $\mathbb{X}$
- There exists an **unknown** parameter $\theta_\star \in \mathbb{R}^d$ such that $f(k) = \langle \phi(k), \theta_\star \rangle$
- For each round t:

1. Compute the posterior mean/covariance on $t-1$ observations

$$\hat{f}_{t-1} = \left(\hat{f}(x)\right)_{x \in \mathbb{X}} \quad \text{and} \quad \hat{\Sigma}_{t-1} = \frac{\sigma^2}{\lambda} \left(k_\lambda(x, x')\right)_{x, x' \in \mathbb{X}}$$

2. Sample a function $\tilde{f} \sim \mathcal{N}_{|\mathbb{X}|}\left(\hat{f}_{t-1}, \hat{\Sigma}_{t-1}\right)$
3. Select action $x_t = \arg\max_{x \in \mathbb{X}} \tilde{f}(x)$
4. Play action x_t
5. Observe reward $r_t = f(x_t) + \epsilon_t$

Summary

- UCB and Thompson Sampling can be extended to exploit action structure
- This allows to consider much larger action spaces
- Kernel-TS is limited to a discrete action space

What about bounds?

Kernel-UCB analysis

$$\mathrm{UCB}_x(t,\lambda,\delta) = \hat{f}_t(x) + \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} \underbrace{\left[\sqrt{\lambda}\|\theta_\star\|_2 + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda)}\right]}_{B(t,\lambda,\delta)}$$

- Minimize regret: $R_T = \sum_{t=1}^{T} (f(x_\star) - f(x_t))$
- Recall: confidence envelope says $|f(x) - \hat{f}_t(x)| \leq \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} B(t,\lambda,\delta)$ simultaneously for all x and t

$$\begin{aligned}
f(x_\star) - f(x_t) &\leq \mathrm{UCB}_{x_\star}(t,\lambda,\delta) - f(x_t) \\
&\leq \mathrm{UCB}_{x_t}(t,\lambda,\delta) - f(x_t) \\
&\leq |\,\mathrm{UCB}_{x_t}(t,\lambda,\delta) - \hat{f}_t(x_t)| + |\hat{f}_t(x_t) - f(x_t)| \\
&\leq 2\sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} B(t,\lambda,\delta)
\end{aligned}$$

Kernel-UCB analysis (cont'd)

$$f(x_\star) - f(x_t) \leq 2\sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} \underbrace{\left[\sqrt{\lambda}\|\theta_\star\|_2 + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda)}\right]}_{B(t,\lambda,\delta)}$$

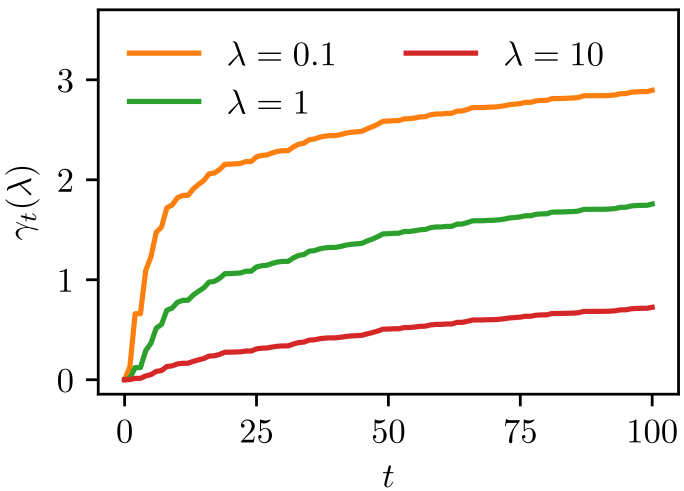
$$\begin{aligned} R_T &= \sum_{t=1}^{T} \left(f(x_\star) - f(x_t)\right) \\ &\leq 2\sum_{t=1}^{T} \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} B(t,\lambda,\delta) \\ &\leq 2B(T,\lambda,\delta) \sum_{t=1}^{T} \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} \end{aligned}$$

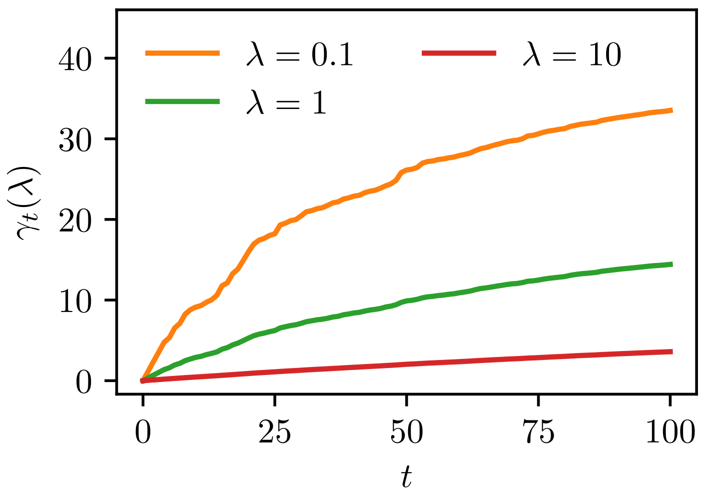
Show that $B(t,\lambda,\delta) \leq B(T,\lambda,\delta)$ for $t \leq T$ and bound the sum!

Kernel-UCB analysis: Bounding $B(t, \lambda, \delta) \leq B(T, \lambda, \delta)$

$$B(t, \lambda, \delta) = \sqrt{\lambda}\|\theta_\star\|_2 + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda)}$$

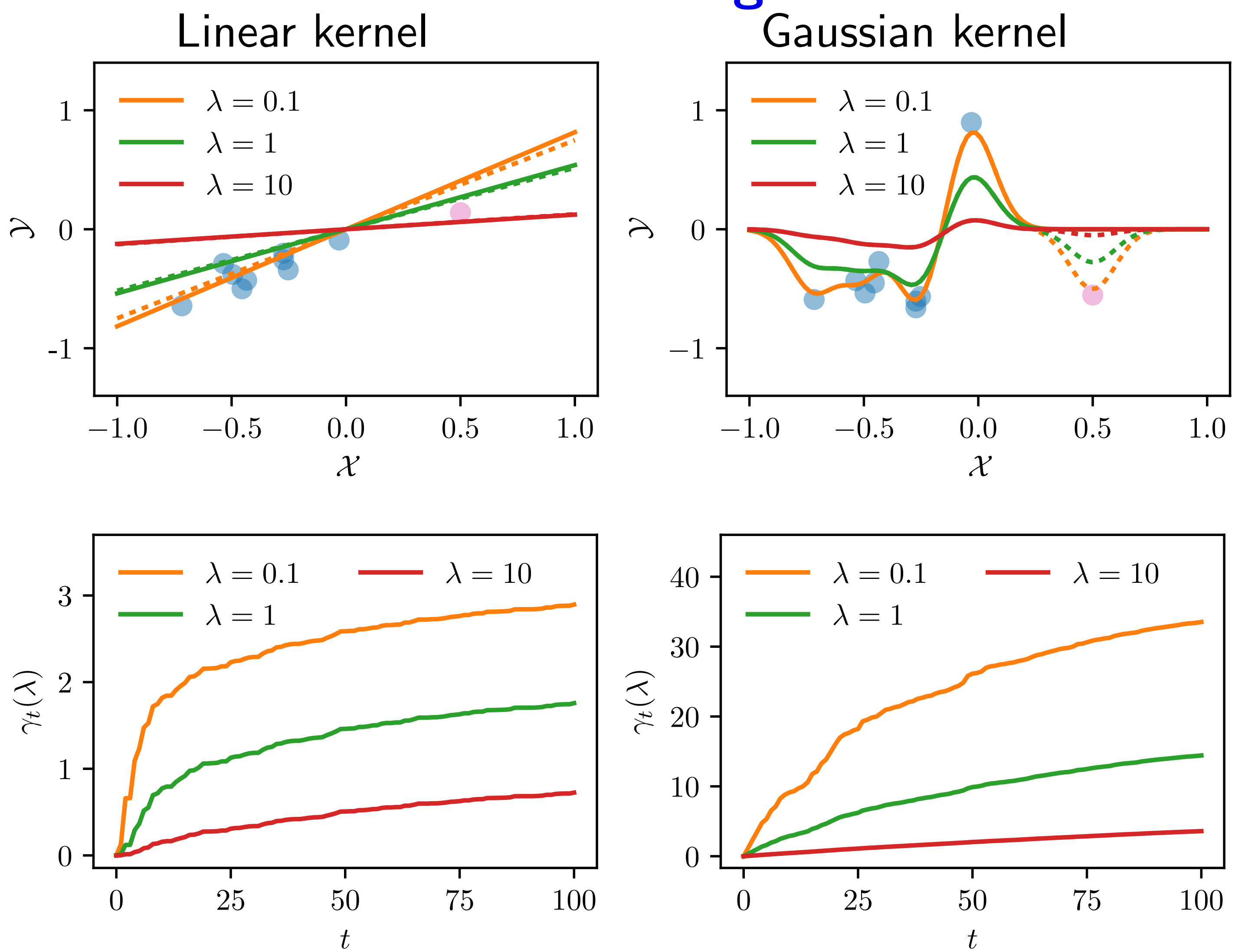
- Recall: Information gain $\gamma_t(\lambda)$ cumulates maximum possible information after t observation. Example: Linear vs Gaussian kernel





This shows that $\gamma_t(\lambda) \leq \gamma_T(\lambda)$ for $t \leq T$

Information gain



Kernel-UCB analysis: Finalizing

Lemma 1.
$$\sum_{t=1}^{T}\sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} \leq \sqrt{T\frac{2}{\lambda\ln(1+1/\lambda)}\gamma_T(\lambda)}$$

$$\begin{aligned}
R_T &\leq 2B(T,\lambda,\delta)\sum_{t=1}^{T}\sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} \\
&\leq 2\left[\sqrt{\lambda}\|\theta_\star\|_2 + \sigma\sqrt{2\ln(1/\delta)+2\gamma_T(\lambda)}\right]\sqrt{T\frac{2}{\lambda\ln(1+1/\lambda)}\gamma_T(\lambda)}
\end{aligned}$$

Impact of kernel? Impact of noise σ? Impact of $\|\theta_\star\|$?

Smoothness of the function

Recall regret:

$$R_T(\pi, \nu) = \sum_{k \in \mathcal{K}} \Delta_k(\nu)\, \mathbb{E}[N_k(T)]$$

- Smooth function: Many suboptimality gaps close to 0

Summary

- We can exploit structure in the action set
- In practice there might be additional information that we can exploit
- Example: Recommendation systems

  What kind of information could we use?

Contextual bandit setting

- Context set $\mathcal{S}$
- Action set $\mathcal{X}$
- On each round t:
  1. Receive context $s_t \in \mathcal{S}$
  2. Select action $x_t \in \mathcal{X}$
  3. Play action x_t
  4. Receive reward $r_t = f(d_t, x_t) + \epsilon_t$
- Goal: Maximize $\sum_{t=1}^{T} f(s_t, x_t)$

Minimize regret: $R_T = \sum_{t=1}^{T} \max_{x \in \mathcal{X}} f(s_t, x) - \sum_{t=1}^{T} f(s_t, x_t)$

What is the optimal action?

Action set perspective

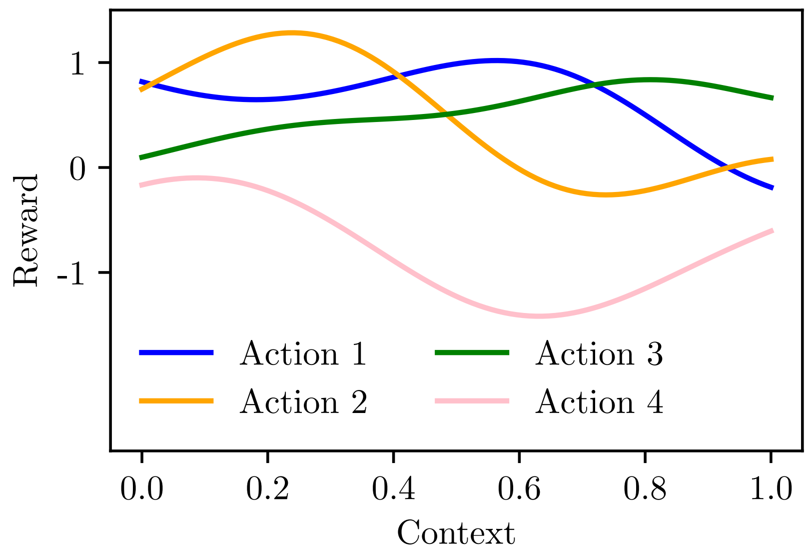
- Augmented action set $\mathcal{A} = \mathcal{S} \times \mathcal{X}$
- On each round t:
  1. Receive available action set $\mathcal{A}_t \subset \mathcal{A}$
  2. Select action $a_t \in \mathcal{A}_t$
  3. Play action a_t
  4. Receive reward $r_t = f(a_t) + \epsilon_t$
- Goal: Maximize $\displaystyle\sum_{t=1}^{T} f(a_t)$

Minimize regret: $\displaystyle R_T = \sum_{t=1}^{T} \max_{a \in \mathcal{A}_t} f(a) - \sum_{t=1}^{T} f(a_t)$

Structured actions!

Specific case: Independent actions

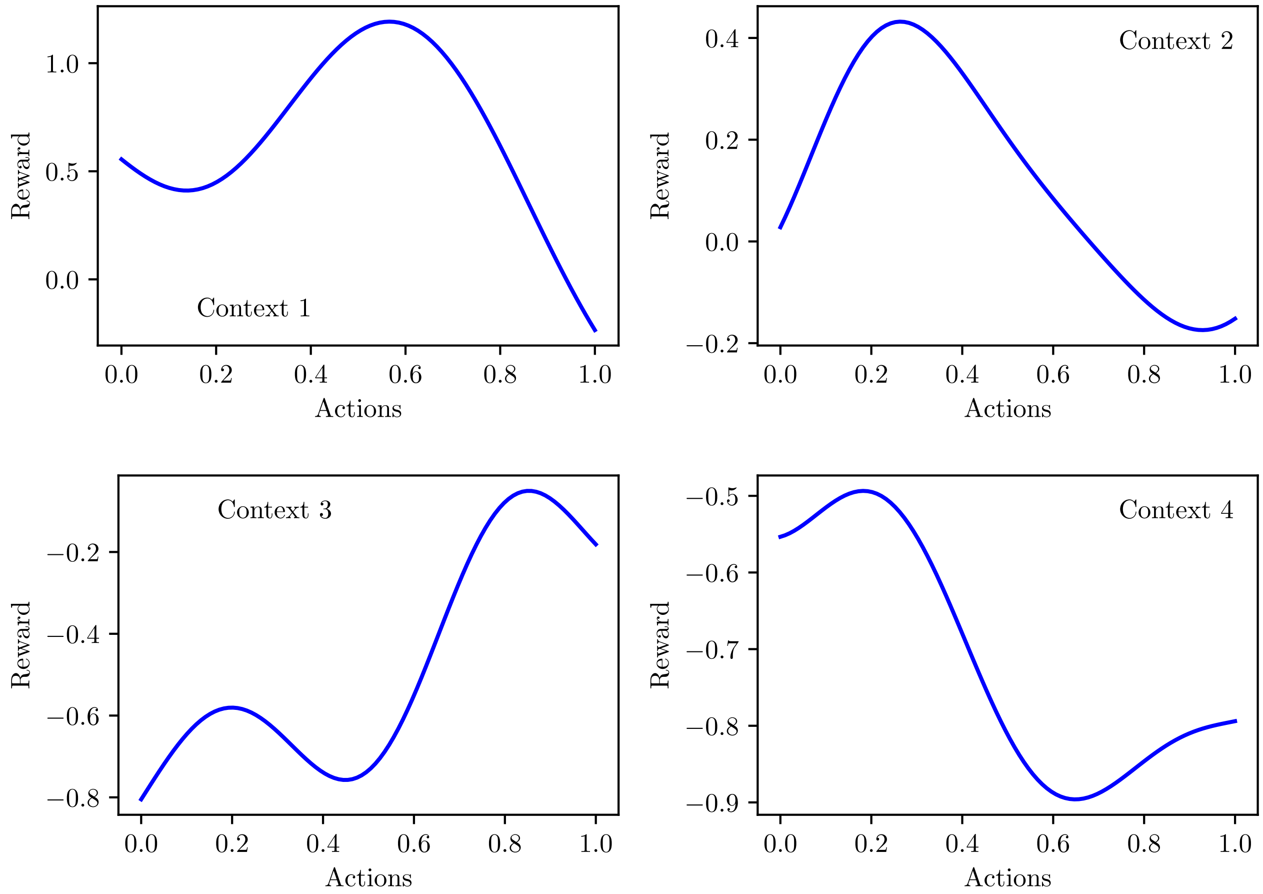
- No information to be shared across actions
- Each action $k \in \mathcal{K}$ has a reward function $f_k : \mathcal{S} \mapsto \mathbb{R}$



The locations that we observe now depend on the context arrival!

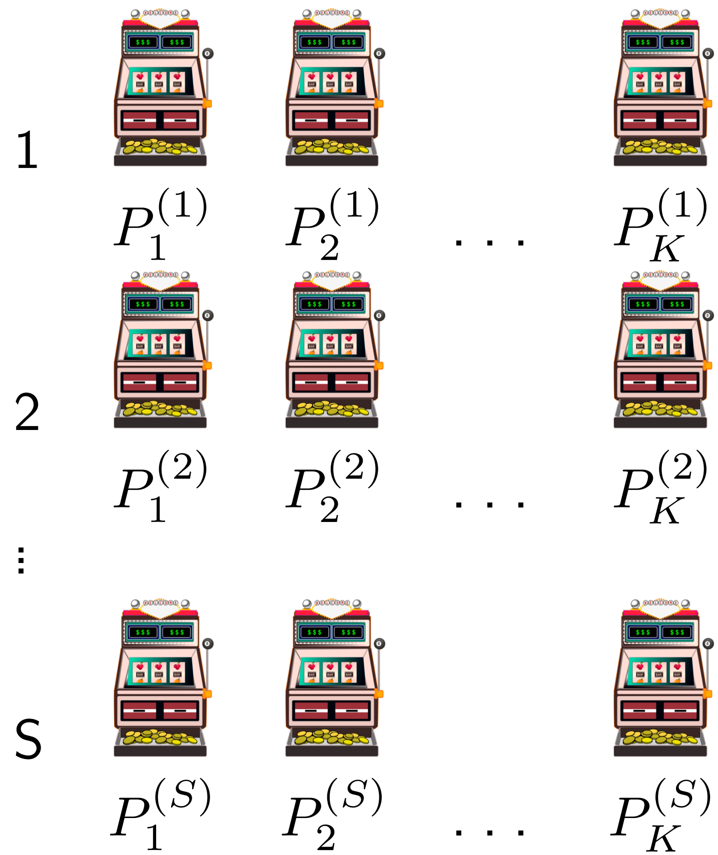
Specific case: Independent contexts

- No information to be shared across contexts
- Each context $s \in \mathcal{S}$ is associated with a reward function $f_s : \mathcal{X} \mapsto \mathbb{R}$



Specific case: Independent actions and contexts

- Each context is an independent stochastic bandit problem



How do you solve this?

How fast can you learn?

Summary

- Results on streaming regression are useful to derive bandits algorithms!
- Structured bandits: quality of estimate depends where you sample
- Contextual bandits: you may not always decide exactly *where* you sample!
- The more information you share the faster you can learn
- This shows up in the information gain