Lecture 18 Online Learning (Part II) and Intro to Reinforcement Learning

Lei Li, **Yu-Xiang Wang**

Recap: Online Learning

- Learning with expert advice
  - A summary of regret bound: # mistakes - Oracle # of mistakes

	Consistency	Halfing	Weighted Majority	Randomized WM
Realizable setting	$\min(T, \lvert\mathcal{H}\rvert)$	$\min(T, \log\lvert\mathcal{H}\rvert)$	$\min(T, \log\lvert\mathcal{H}\rvert)$	$\min(T, \log\lvert\mathcal{H}\rvert)$
Agnostic setting	n.a.	n.a.	$(1+\epsilon)m + \log\lvert\mathcal{H}\rvert/\epsilon$	$\sqrt{m \log\lvert\mathcal{H}\rvert} = O(\sqrt{T \log\lvert\mathcal{H}\rvert})$

Recap: Hedge (aka Exponential weighted average) algorithm

EXPONENTIAL-WEIGHTED-AVERAGE (N)

for $i \leftarrow 1$ **to** N **do**

  $w_{1,i} \leftarrow 1$

for $t \leftarrow 1$ **to** T **do**

  RECEIVE(x_t)

  $\widehat{y}_t \leftarrow \frac{\sum_{i=1}^{N} w_{t,i} y_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$

  RECEIVE(y_t)

  for $i \leftarrow 1$ **to** N **do**

    $w_{t+1,i} \leftarrow w_{t,i}\, e^{-\eta L(\widehat{y}_{t,i}, y_t)}$

return $\mathbf{w}_{T+1}$

- Works for linear loss function in its first argument
  - That is bounded
- Also works for any convex loss function in its first argument
  - Why?

This lecture

- Online Learning (Part II)
    - Online Gradient Descent

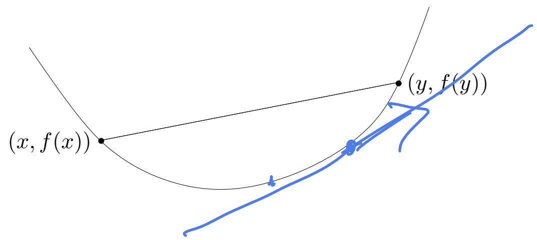
- Reinforcement Learning
    - Problem setup
    - Markov Decision Processes

Recap: Convex functions / sets and subgradient

Convex function: $f : \mathbb{R}^n \to \mathbb{R}$ such that $\mathrm{dom}(f) \subseteq \mathbb{R}^n$ convex, and

$$f(tx + (1-t)y) \le tf(x) + (1-t)f(y) \quad \text{for all } 0 \le t \le 1$$

and all $x, y \in \mathrm{dom}(f)$



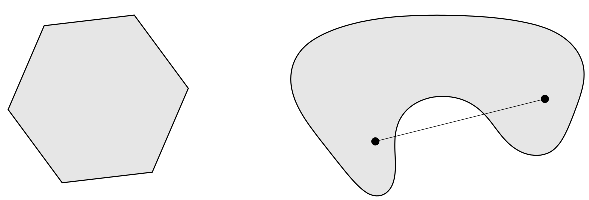
- First order definition / subgradient

$$f(y) \ge f(x) + g_x^T (y - x) \quad \forall x, y \iff f \text{ is convex}$$

g subgradient of f at x

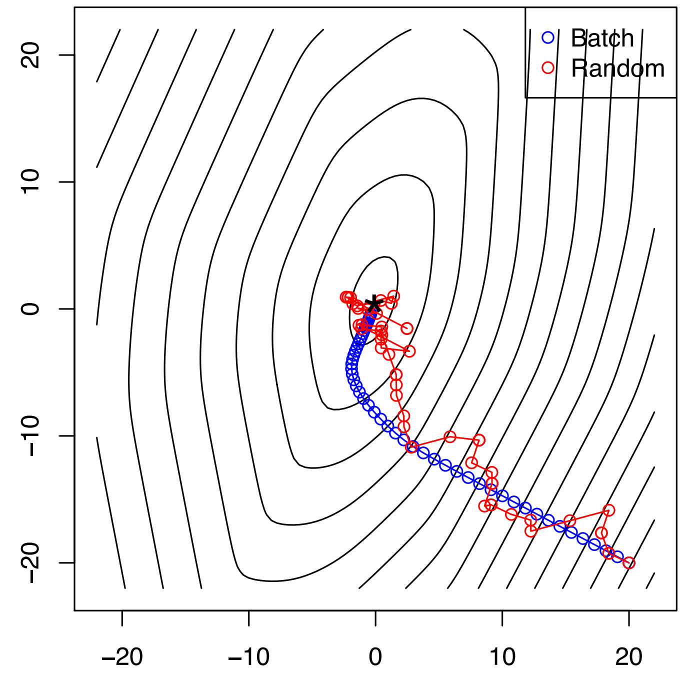
Convex set: $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \implies tx + (1-t)y \in C \quad \text{for all } 0 \le t \le 1$$



Recap: Gradient Descent and SGD (from Lecture 4)

- Problem: $\min_{\theta} f(\theta)$
- GD alg.: $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t)$
- SGD alg.: $\theta_{t+1} = \theta_t - \eta_t \hat{\nabla} f(\theta_t)$



- Example when solving ERM:

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, (x_i, y_i))$$

- Pick a **single** data point i uniformly at random

$$\nabla_\theta \ell(\theta, (x_i, y_i))$$

(Projected) Subgradient "Descent"

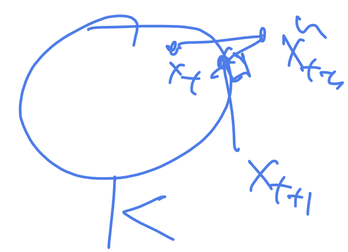
- When we have constraints and non-differentiable convex functions, we can use
  - Projected Subgradient method
  - Projected Stochastic subgradient method

g_t is a subgradient of f at x_t

$\min_{x \in K} f(x)$

$\tilde{x}_{t+1} = x_t - \eta_t \cdot g_t$

$x_{t+1} = \Pi_K(\tilde{x}_{t+1}) = \arg\min_{x \in K} \|x - \tilde{x}_{t+1}\|_2^2$



Convex f

Convergence $O\left(\frac{1}{\sqrt{t}}\right)$

$x_{t+1} = \Pi_K(x_t - \eta_t \hat{g}_t)$

$E[\hat{g}_t] = g_t$

$|E[\|\hat{g}_t - g_t\|_2^2]| = O(1)$

The problem of Online Convex Optimization

- Problem setup:

for $t = 1, 2, \cdots, T$

learner plays $x_t \in K$

nature chooses f_t

learner incurs a loss $f_t(x_t)$

- Performance metric --- regret

$$\text{Regret}_T = \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(u)$$

$$u = \arg\min_{u \in K} \sum f_t(u)$$

Examples of OCO problems

- Example 1: Prediction with Expert Advice

$$f_t = \langle \ell_t, x_t \rangle \quad \|\ell_t\|_\infty \le 1$$

$$\mathcal{K} = \Delta^{n-1}$$

- Example 2: Online Linear models

$$f_t(x) = \ell_t(x^T \phi_t)$$

ϕ_t: feature vector $\in \mathbb{R}^d$

$$(x^T \phi_t - y_t)^2$$

$$\log(1 + \exp(-y_t \cdot x^T \phi_t))$$

$y_t \in \{-1, 1\}$

- Example 3: Portfolio Selection

$$f_t(x) = -\log(r_t^T x)$$

$$r_t = \begin{pmatrix} 1.1 \\ 0.9 \\ 1.4 \\ 0.6 \end{pmatrix}$$

$$= -\log(r_t^T x) + \log(\mathbf{1}^T x)$$

$\mathbf{1}^T x = 0$

Algorithm: OGD, i.e., Online (projected) (sub)Gradient Descent

- Standard projected subgradient updates

$$x_{t+1} = \Pi_{K}\left(x_t - \eta_t \nabla f_t(x_t)\right), \quad \nabla f_t(x_t) = g_{t,x_t}$$

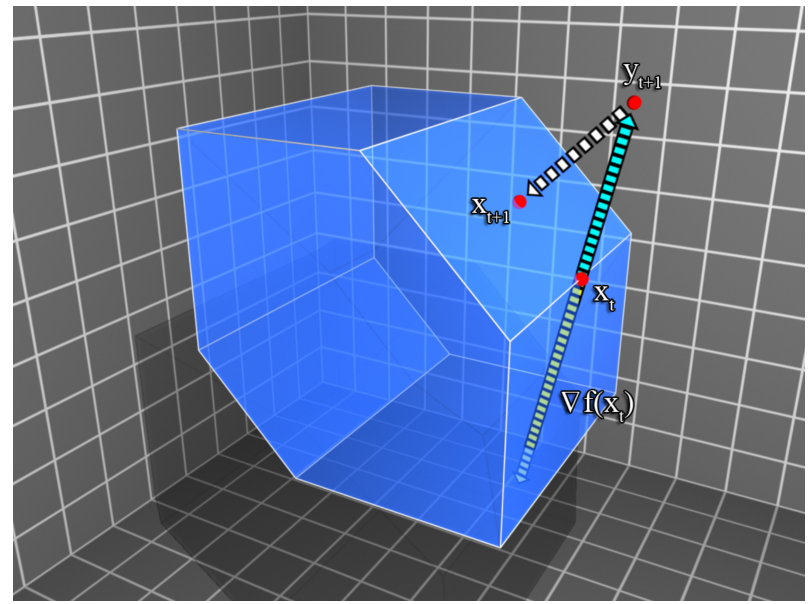


Figure 3.1: Online gradient descent: the iterate $\mathbf{x}_{t+1}$ is derived by advancing $\mathbf{x}_t$ in the direction of the current gradient ∇_t, and projecting back into $\mathcal{K}$.

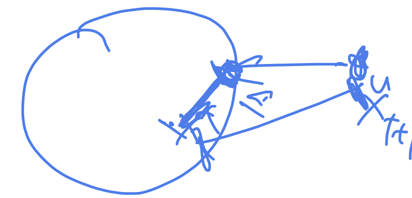
- Assumptions needed:
  - Bounded domain

    $\|x - x'\|_2 \le D \quad \forall x, x' \in K$

  - Lipschitz loss functions

    f_t is G-Lipschitz $\quad \forall x, x' \; |f_t(x) - f_t(x')| \le G \|x - x'\|_2$

    $\|g_x\|_2 \le G \quad \forall x$



Analysis of OGD

- By convex functions

x^*

$$f_t(x_t) - f_t(x^*) \le g_t^T(x_t - x^*)$$

$$f_t(x^*) \ge f_t(x_t) + g_t^T(x^* - x_t)$$

- By the update rule (and property of projection)

$$\|x_{t+1} - x^*\|^2 = \|\Pi_K(x_t - \eta_t g_t) - x^*\|_2^2 \le \|x_t - \eta_t g_t - x^*\|^2 = \|x_t - x^*\|^2 + \eta_t^2\|g_t\|^2 - 2\eta_t g_t^T(x_t - x^*)$$

- Put them together!

$$f_t(x_t) - f_t(x^*) \le g_t^T(x_t - x^*) \le \frac{1}{2\eta_t}\left(\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right) + \frac{\eta_t}{2}\|g_t\|_2^2$$

Analysis of OGD (continues)

- Telescoping

$$\sum_t f_t(x_t) - f_t(x^*) \le \frac{1}{2\eta}\left(\underbrace{\|x_1 - x^*\|^2}_{\le D^2} - \underbrace{\|x_{T+1} - x^*\|^2}_{\le 0}\right) + \sum_{t=1}^{T} \frac{\eta}{2} \underbrace{\|g_t\|^2}_{\le G^2}$$

$$\le \frac{1}{2\eta} D^2 + \frac{T\eta}{2} G^2$$

$$= D \cdot G \sqrt{T} = o(T)$$

$$\frac{x^2 + y^2}{2} \ge xy$$

Regret bound for OGD

Theorem 3.1. Online gradient descent with step sizes $\{\eta_t = \frac{D}{G\sqrt{t}},\ t \in [T]\}$ guarantees the following for all $T \geq 1$:

$$\text{regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^\star \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^\star) \ \leq \frac{3}{2} GD\sqrt{T}$$

- "Any-time" algorithm with a decreasing learning rate schedule

- Learning rate depends on t. (exercise to prove that this works.)

Online to Batch conversion: How do I use OGD to solve ERM?

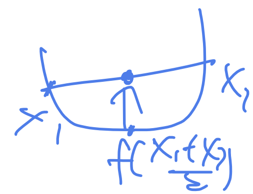
$f_t \overset{iid}{\sim} P$

$\mathbb{E}_P[f_1] = f$

$\min_x f(x) \leftarrow$ Risk

$\frac{1}{T}\sum_{t=1}^{T} f_t(x)$ Empirical Risk

$\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t$



convex f

$\mathbb{E}[f(\bar{x})] - f(x^*) \le \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f(x_t)\right] - f^*$

$\mathbb{E}[f_t] = f$

$\le \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f(x_t)\right] - f(x^*)$

$f(x^*) \ge f(x_t) + \nabla f(x_t)^T(x^* - x_t)$

$\le \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \nabla f(x_t)^T(x_t - x^*)\right]$

$\mathbb{E}\, g_t = \mathbb{E}\nabla f_t = \nabla f$

$f_t \sim P$

$= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} g_t^T(x_t - x^*)\right]$

$\mathbb{E}[f_1] = f$

$\le \frac{1}{T}\mathbb{E}\left[\sum_t f_t(x_t) - f_t(x^*)\right] = \frac{1}{T}\text{Regret} = O\left(\frac{GD}{\sqrt{T}}\right)$

Regret

Checkpoint

- Online learning
  - Operates in an adversarial environment
  - Almost no assumptions. Do not even use probability theory

- The idea of regret and no-regret learning algorithms

- Useful algorithmic ideas:
  - Hedge / Exponential Weighted Averages
  - Online gradient descent

What we did not cover

- The strongly convex case
- Adapting to the geometry
  - AdaGrad / ADAM
- Adaptive regret / dynamic regret

- Modern applications to Ensemble learning, AutoML

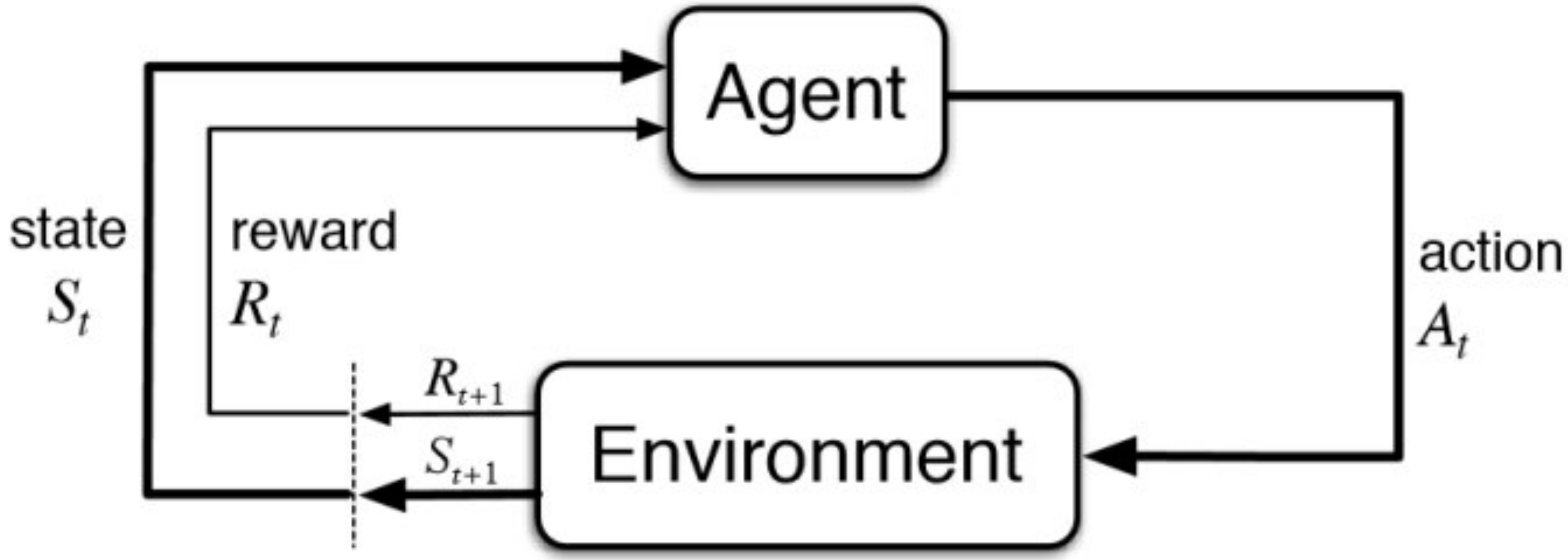
[PDF] Hyperband: A novel bandit-based approach to hyperparameter optimization
L Li, K Jamieson, G DeSalvo, A Rostamizadeh… - The Journal of Machine …, 2017 - jmlr.org
Performance of machine learning algorithms depends critically on identifying a good set of hyperparameters. While recent approaches use Bayesian optimization to adaptively select …
Save Cite Cited by 1612 Related articles Import into BibTeX

This lecture

- Online Learning (Part II)
  - Online Gradient Descent

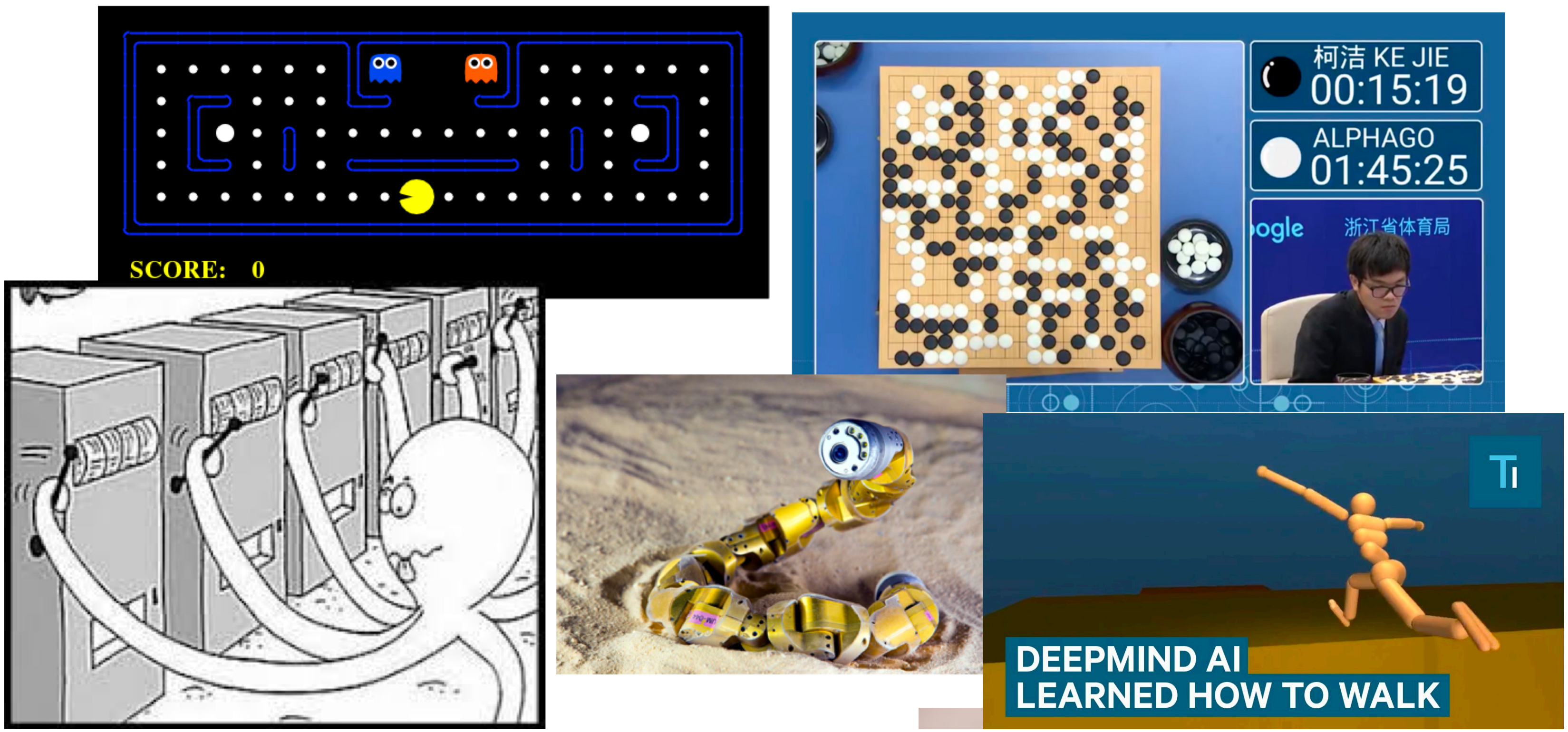
- Reinforcement Learning
  - Problem setup
  - Markov Decision Processes

An RL agent learns interactively through the feedbacks of an environment.



- Learning how the world works (dynamics) and how to maximize the long-term reward (control) at the same time.

Reinforcement learning is among the hottest area of research in ML!



“RL” is Top 1 Keyword at NeurIPS’2021, appearing 199 times
“Deep Learning” only 129 times [source]

Applications of RL in the real life

- RL for robotics.
- RL for dialogue systems.
- RL for personalized medicine.
- RL for self-driving cars.
- RL for new material discovery.
- RL for sustainable energy.
- RL for feature-based dynamic pricing.
- RL for maximizing user satisfaction.
- RL for QoE optimization in networking
- …

Reinforcement learning problem setup

- State, Action, Reward and Observation

$$S_t \in \mathcal{S} \quad A_t \in \mathcal{A} \quad R_t \in \mathbb{R} \quad O_t \in \mathcal{O}$$

- Policy: $\pi : \mathcal{S} \to \mathcal{A}$
  - When the state is observable:
  - Or when the state is not observable

$$\pi_t : (\mathcal{O} \times \mathcal{A} \times \mathbb{R})^{t-1} \to \mathcal{A}$$

- Learn the best policy that maximizes the expected reward
  - Finite horizon (episodic) RL: $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}[\sum_{t=1}^{T} R_t]$

T: horizon

  - Infinite horizon RL:

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} R_t]$$

γ: discount factor

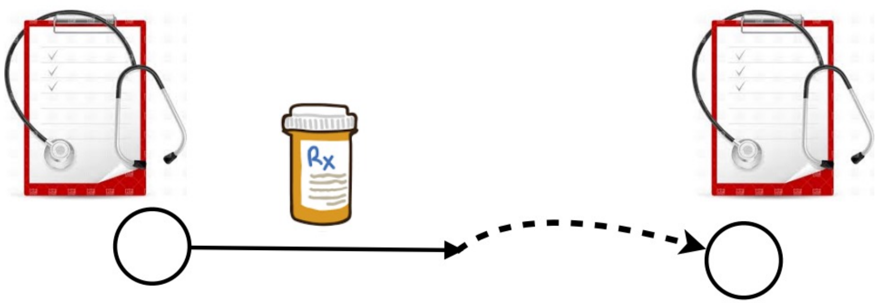
RL for robot control

- States: The physical world, e.g., location/speed/acceleration and so on.
- Observations: camera images, joint angles
- Actions: joint torques
- Rewards: stay balanced, navigate to target locations, serve and protect humans, etc.

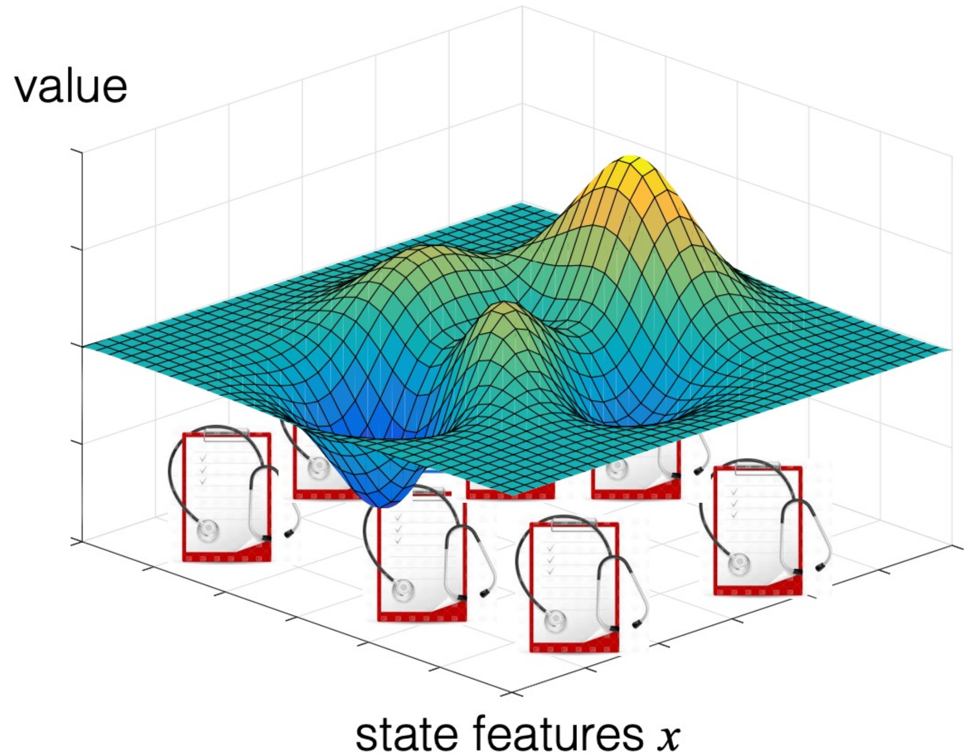
RL for Inventory Management

- State: Inventory level, customer demand, competitor's inventory
- Observations: current inventory levels and sales history
- Actions: amount of each item to purchase
- Rewards: profit

RL for Adaptive medical treatment



- State: diagnosis
- Action: treatment
- Reward: progress in recovery



(example / illustration due to Nan Jiang)

Example: Supervised learning vs RL in movie recommendation

- Bob is described by a feature vector
  - s =[Previous movies watched / Rating / Written reviews]

- Supervised learning predicts how likely Bob will click on "aliens vs predators"

- Reinforcement learning aims at controlling Bob
  - So in the future, Bob will develop a taste for "aliens vs predators" (e.g., from having watched "aliens" and "predators" both).