Reinforcement Learning

Machine Learning and Optimization

Marek Petrik

4/20/2017

Branches of Machine Learning

- Supervised Learning

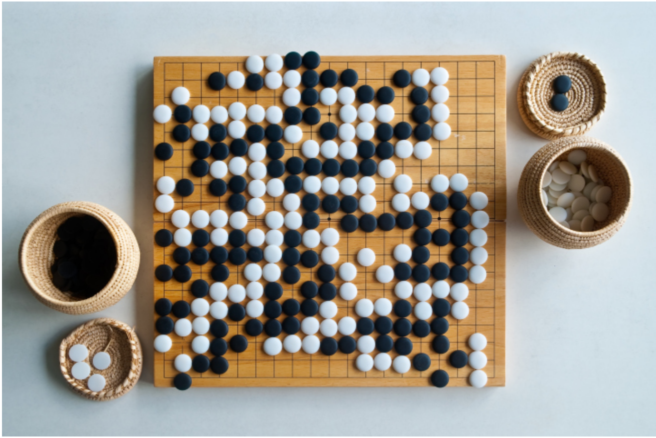
Branches of Machine Learning

- Supervised Learning
- Unsupervised Learning

Branches of Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning (maybe): Machine learning + decisions

AlphaGo: Computers Beat Humans in Go



Photograph by Saran Poroong—Getty Images/iStockphoto

Wumpus World

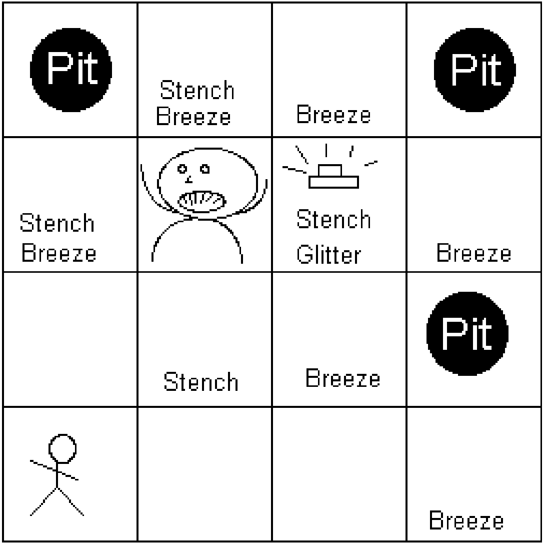
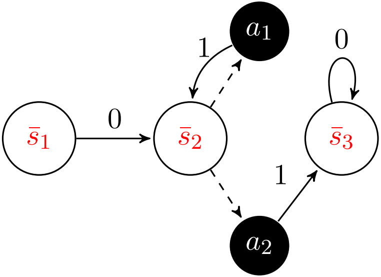


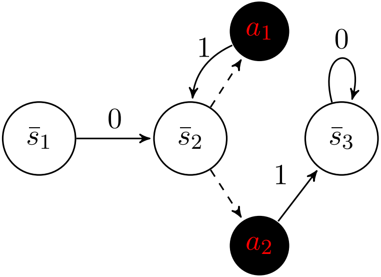
Figure 1

Markov Decision Process



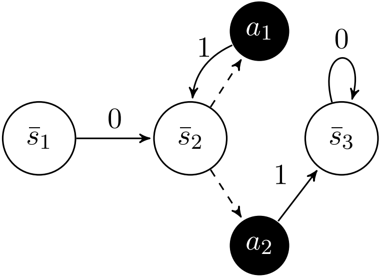
- States

Markov Decision Process



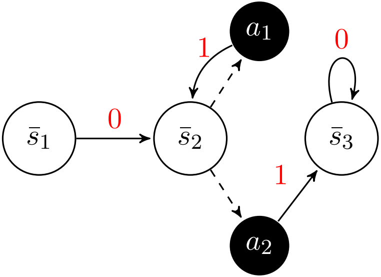
- Actions

Markov Decision Process



- Transition probabilities: P

Markov Decision Process



- Rewards: r

MDP Objective: Discounted Infinite Horizon

Solution

Policy π maps *states* $\rightarrow$ *actions*

MDP Objective: Discounted Infinite Horizon

Solution

Policy π maps *states* $\rightarrow$ *actions*

Return for discount factor: $\gamma \in [0, 1]$

$$\rho(\pi) = \mathbf{E}_\alpha \left[\sum_{t=0}^{\infty} \gamma^t \, \text{reward}_t \right]$$

MDP Objective: Discounted Infinite Horizon

Solution

Policy π maps *states* → *actions*

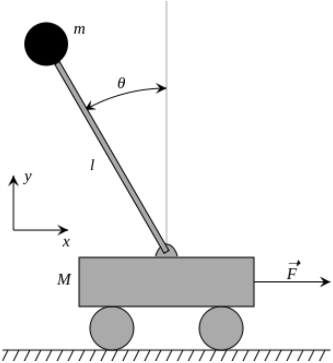
Return for discount factor: $\gamma \in [0, 1]$

$$\rho(\pi) = \mathbf{E}_\alpha \left[\sum_{t=0}^{\infty} \gamma^t \, \text{reward}_t \right]$$

Optimal policy

$$\pi^\star \in \arg\max_{\pi} \; \rho(\pi)$$

Balancing Inverted Pendulum



- Balance a ball on top of the pole
- Can apply force on the cart
- Uncertainty in magnitude of force
- Decide when and how much force to apply

Energy Storage

- Decide how much to charge and discharge
- Based on stochastic energy prices
- **Solution**: Policy:
  - Buy low and sell high

Energy Storage

- Decide how much to charge and discharge
- Based on stochastic energy prices
- **Solution**: Policy:
  - Buy low and sell high
  - But how much?

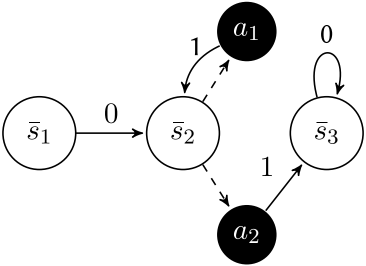
MDP Models

- Energy storage
  - States: Battery charge level, capacity, energy price
  - Actions: Charge or discharge the battery
  - Transitions: Battery dynamics and stochastic energy price
  - Reward: Money earned

MDP Models

- Inverted pendulum
    - States: Angle and velocity of pendulum
    - Actions: Magnitude and direction of force
    - Transitions: Pendulum dynamics (differential equations)
    - Reward: -1 when falls 0 otherwise

Optimal Solution

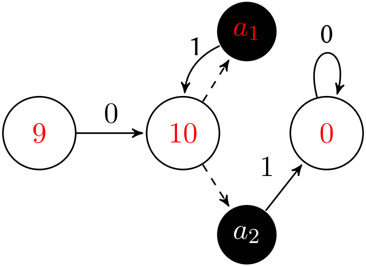


$\gamma = 0.9$

Value Function of π

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi_{s,a} \Big(r_a(s) + \gamma \sum_{s' \in \mathcal{S}} P_a(s, s') v_\pi(s') \Big)$$

Optimal Solution

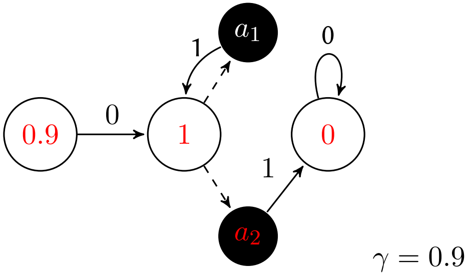


$\gamma = 0.9$

Value Function of π

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi_{s,a} \Big(r_a(s) + \gamma \sum_{s' \in \mathcal{S}} P_a(s, s') v_\pi(s') \Big)$$

Optimal Solution



Value Function of π

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi_{s,a} \Big(r_a(s) + \gamma \sum_{s' \in \mathcal{S}} P_a(s, s') v_\pi(s') \Big)$$

Optimal Solution

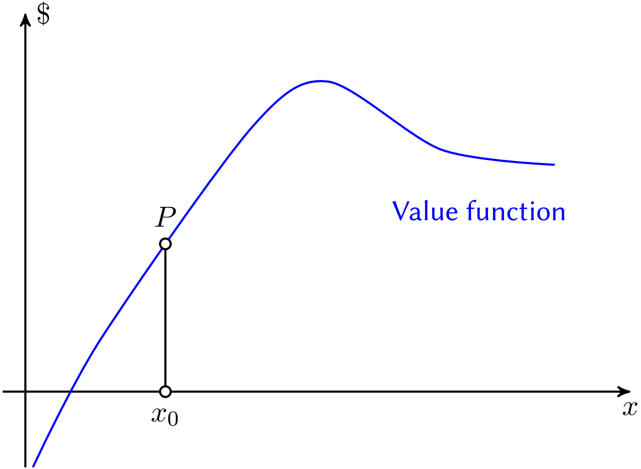
Value Function of π

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi_{s,a}\Big(r_a(s) + \gamma \sum_{s'\in\mathcal{S}} P_a(s,s')v_\pi(s')\Big)$$

Bellman Optimality

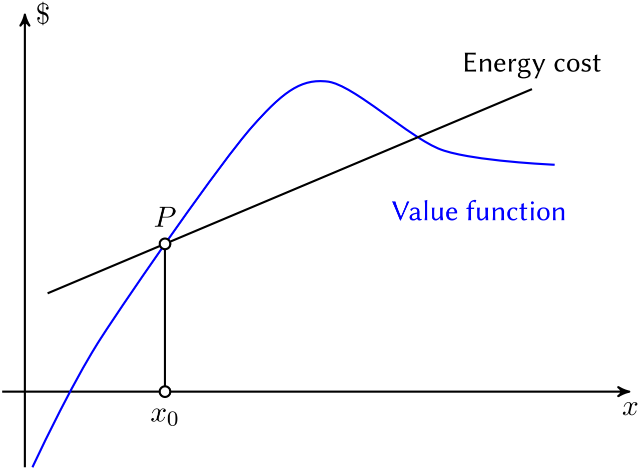
$$v^\star(s) = \max_{\pi\in\Pi_R} \sum_{a\in\mathcal{A}_s} \pi_{s,a}\left(r_a(s) + \gamma \sum_{s'\in\mathcal{S}} P_a(s,s')\,v^\star(s')\right).$$

Energy Storage Value Function: Low Energy Cost



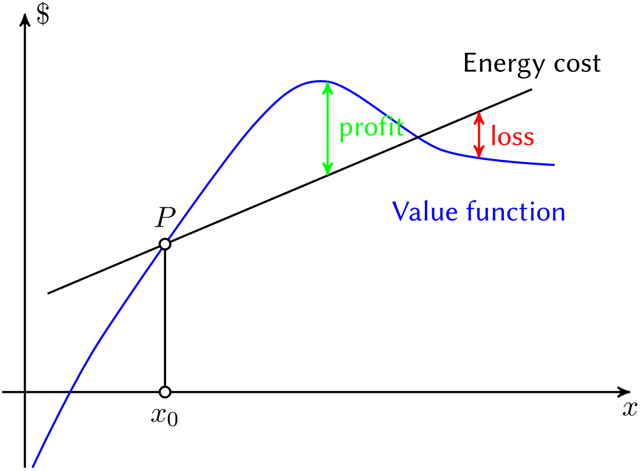
- x_0 – current battery charge

Energy Storage Value Function: Low Energy Cost



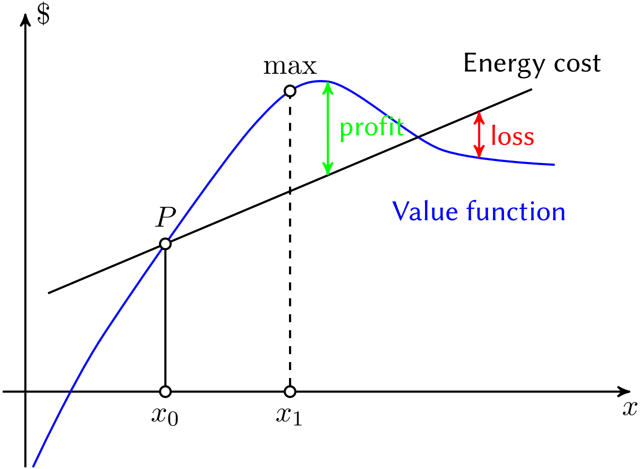
- x_0 – current battery charge

Energy Storage Value Function: Low Energy Cost



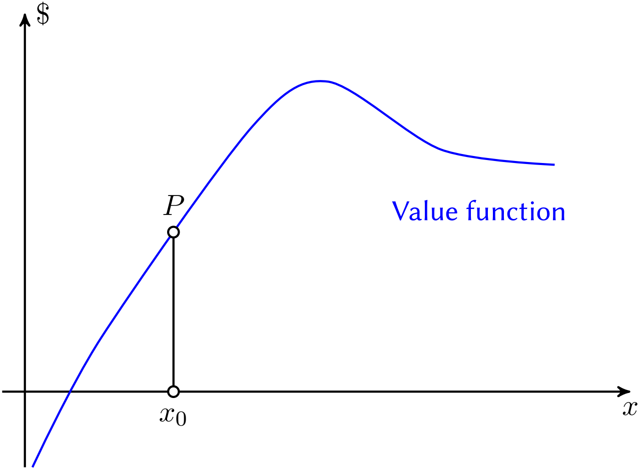
- x_0 – current battery charge

Energy Storage Value Function: Low Energy Cost



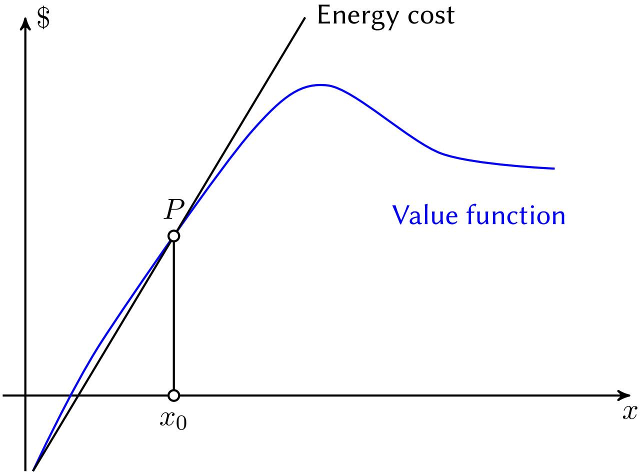
- x_0 – current battery charge
- x_1 – next battery charge

Energy Storage Value Function: High Energy Cost



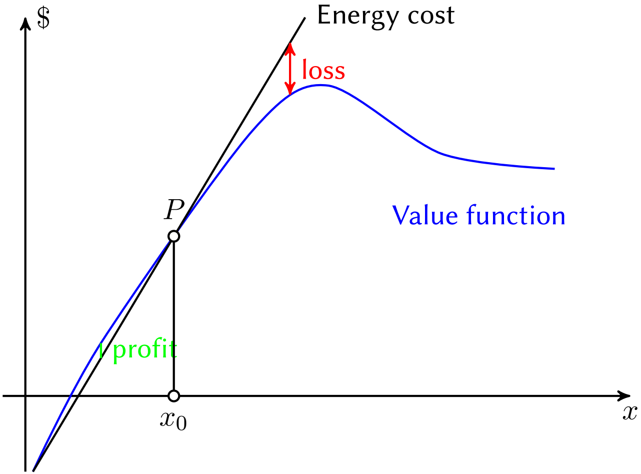
- x_0 – current battery charge

Energy Storage Value Function: High Energy Cost



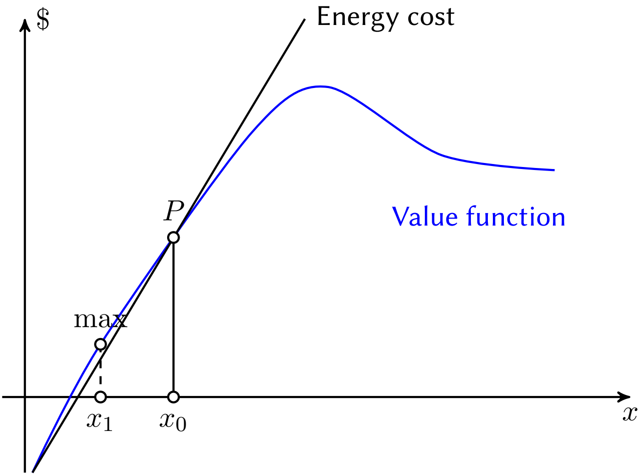
- x_0 – current battery charge

Energy Storage Value Function: High Energy Cost



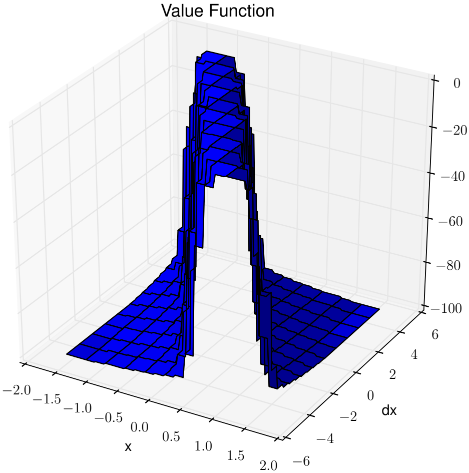
- x_0 – current battery charge

Energy Storage Value Function: High Energy Cost



- x_0 – current battery charge
- x_1 – next battery charge

Pendulum Value Function



Reinforcement learning

- Solve large MDPs using only historical data:
  - Rewards and transition probabilities are not known
  - Can interact with the environment and observe outcomes and rewards
  - There are too many states, the solution must generalize (Machine learning)
- How much to explore and exploit (Multi-armed bandits)

Reinforcement learning

- Solve large MDPs using only historical data:
  - Rewards and transition probabilities are not known
  - Can interact with the environment and observe outcomes and rewards
  - There are too many states, the solution must generalize (Machine learning)
- How much to explore and exploit (Multi-armed bandits)
- **Want to learn more?**: Come to my CS 980: Advanced ML.