Preference-Based Reinforcement Learning: A Policy Iteration Algorithm

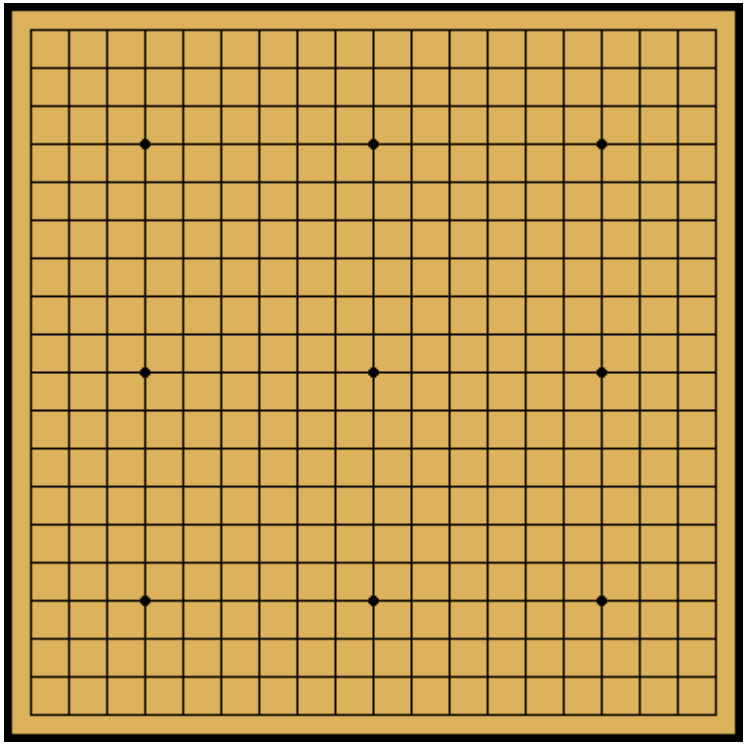
Weiwei Cheng

a joint work with
Johannes Fürnkranz, Eyke Hüllermeier, and Sang-Hyeun Park

TU Darmstadt

Marburg University

Some Terminologies



Reinforcement learning is to learn what to do – how to map situations to actions – so as to maximize a reward signal.

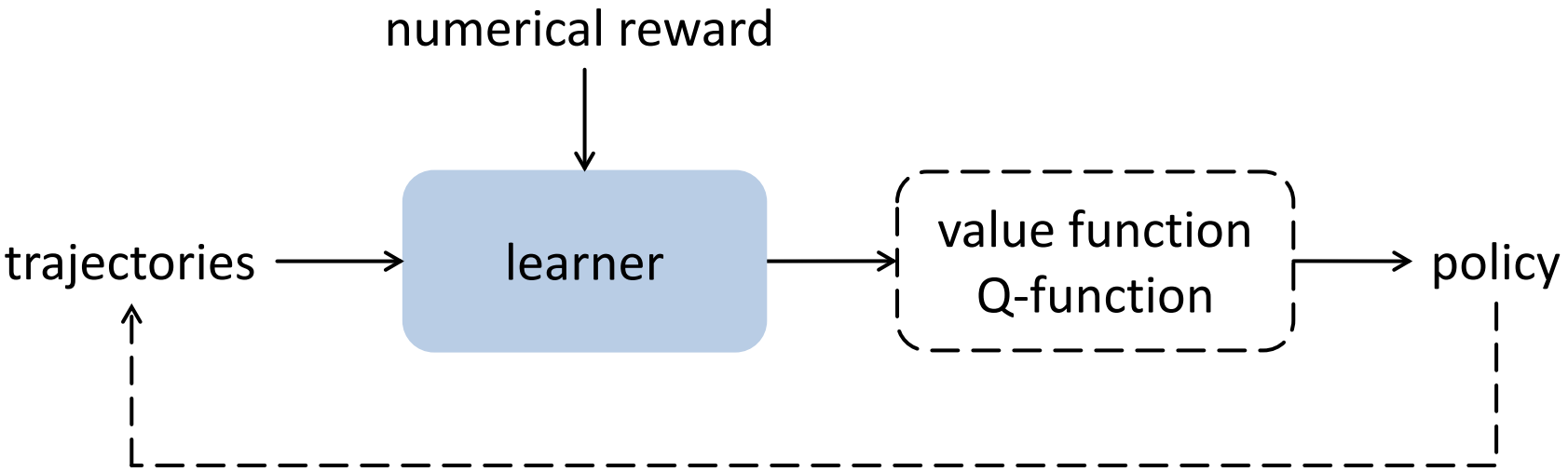
policy: a mapping from states to actions

reward: the consequence of actions in given states

value function: the amount of reward an agent can expect to accumulate over the future, starting from a given state

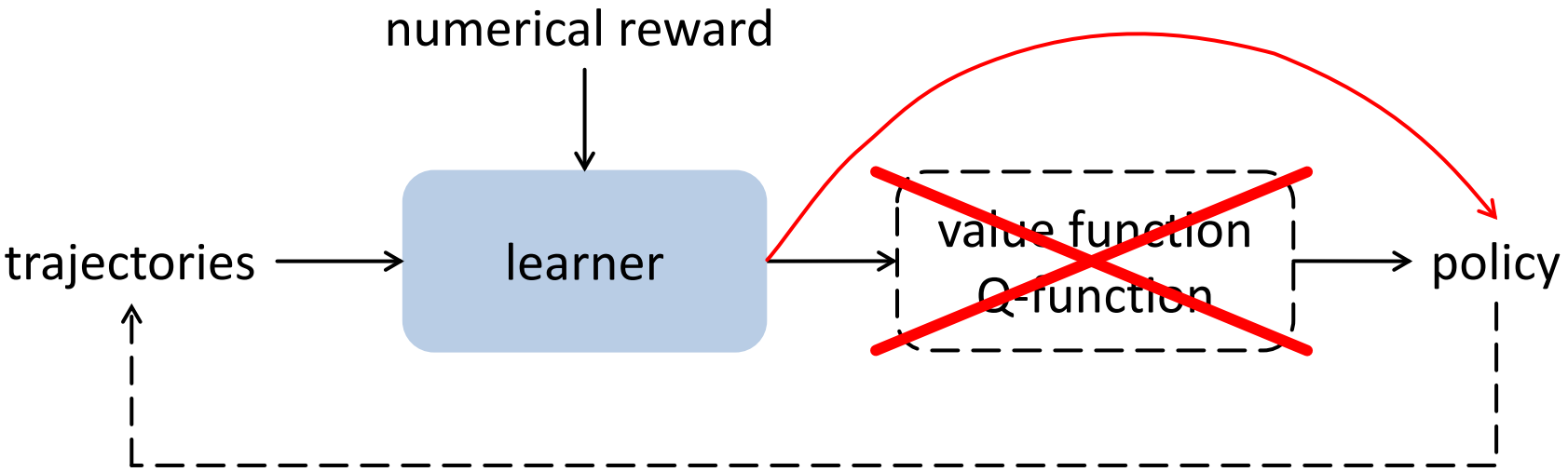
trajectories: the state-action sequences

Conventional Reinforcement Learning



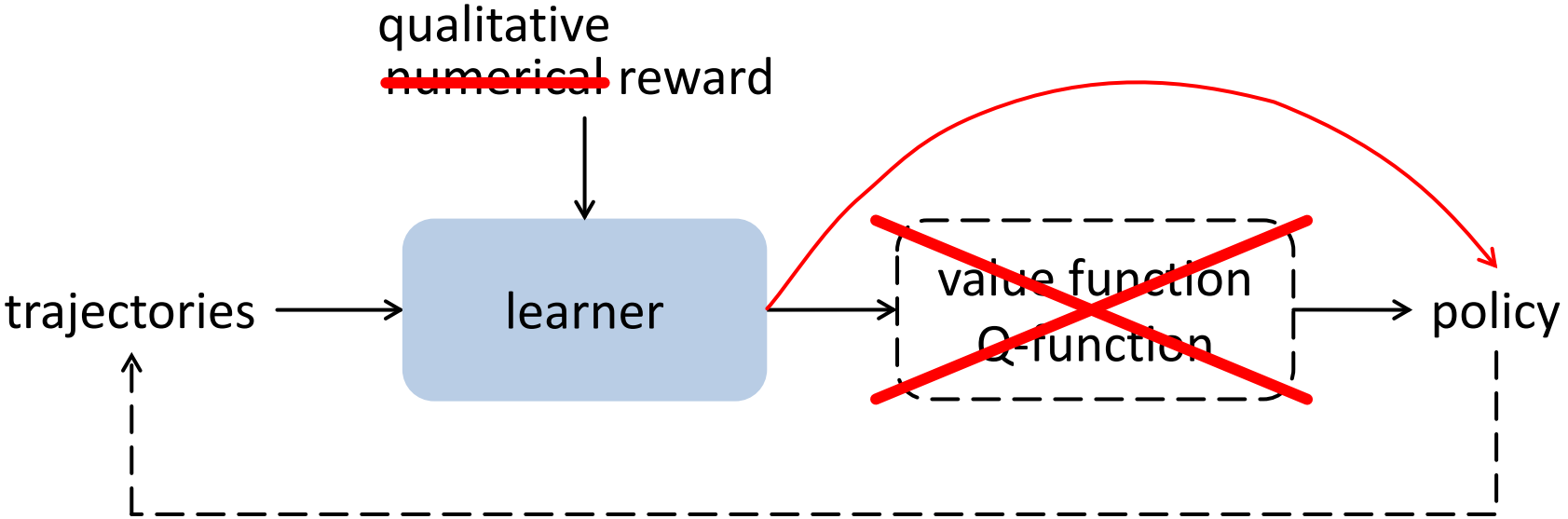
- The learner produces a function that estimates the value of states or state/action pairs: e.g., Q-learning, TD(λ), etc.
- The policy uses this function for taking actions: e.g., greedy, ε-greedy policies, etc.

Policy Learning



- The learner directly learns a policy:
  - actor-critic methods learn both a value function and a policy
  - policy gradient methods search in the space of parametrized policies
  - e.g., a policy is a linear function that maps a state to continuous actions
- Estimation of expected reward may not be necessary!

Preference-Based Reinforcement Learning

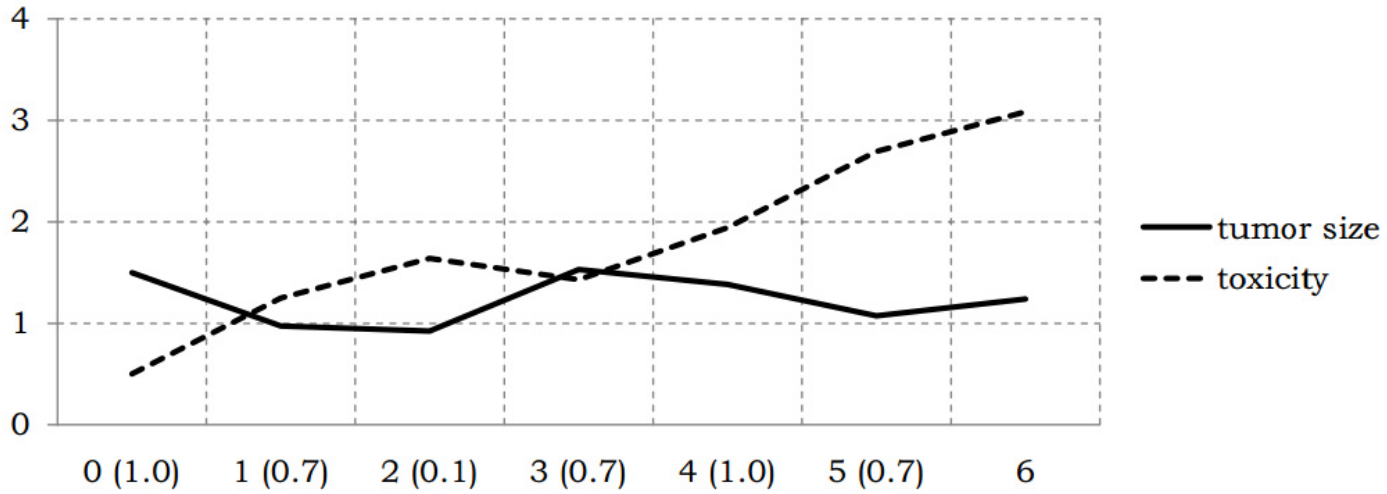


- Training information:
  - preferences over trajectories, policies, or states and actions
- Preference-based policy learning:
  - the policy function is a label ranker that ranks all actions in a given state
  - we know the order of actions but not their valuation

Example: Cancer Clinical Trials

- A simulation model of optimal therapy design in cancer treatment proposed in Zhao et al. 2009
- The model captures a number of essential factors in cancer treatment with chemotherapy.

Example: Cancer Clinical Trials



- The two state variables, the tumor size and the toxicity, are modeled using a system of recurrence relation. They are depended on the action (the dosage) taken at each state.
- The possible death of a patient in the course of a treatment is modeled by a hazard rate model.

Example: Cancer Clinical Trials

We decompose this reward function R_t into three parts: $R_{t,1}(D_t, W_{t+1}, M_{t+1})$ due to survival status, $R_{t,2}(W_t, D_t, W_{t+1})$ due to wellness effects, and $R_{t,3}(M_t, D_t, M_{t+1})$ due to tumor size effects. It can be described by

$$R_{t,1}(D_t, W_{t+1}, M_{t+1}) = -60 \quad \text{if patient died}$$

otherwise,

$$R_{t,2}(W_t, D_t, W_{t+1}) = \begin{cases} 5 & \text{if } W_{t+1} - W_t \leqslant -0.5 \\ -5 & \text{if } W_{t+1} - W_t \geqslant 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$R_{t,3}(M_t, D_t, M_{t+1}) = \begin{cases} 15 & \text{if } M_{t+1} = 0 \\ 5 & \text{if } M_{t+1} - M_t \leqslant -0.5, \text{ but } M_{t+1} \neq 0 \\ -5 & \text{if } M_{t+1} - M_t \geqslant 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Statist. Med. 2009; **28**:3294–3315
DOI: 10.1002/sim

Example: Cancer Clinical Trials

- Such a numerical reward system must be defined with the help of domain knowledge.
- Especially, the aggregation of the criteria (tumor size, toxicity, death) is very hard.
- Our approach proceeds from a preference relation on trajectories $\sigma = (s_0, a_0, s_1, \dots, s_{n-1}, a_{n-1}, s_n)$:

  We say $\sigma \succ \sigma'$, if
  - the patient survives under σ but not σ' or
  - $C_X \leq C_X'$ and $C_Y \leq C_Y'$ in case patient survives both σ and σ',

  where C_X is the maximal toxicity during the treatment and C_Y is the tumor size at the end of the therapy.

The relation $\succ$ is a partial order!

Approximate Policy Iteration with Roll-Outs

(Lagoudakis & Parr, ICML03)

Assuming

- a generative model of the underlying Markov process, and
- with this model actions and rewards can be sampled,

so that we can perform roll-outs (generate trajectories)

Roll-out

- estimate the value $Q^{\pi}(s, a)$ for performing action a at state s and following policy π thereafter
- by performing the action and then repeatedly following the policy for at most T steps
- and returning the average of the observed rewards.

These roll-outs are used for training a policy…

Approximate Policy Iteration with Roll-Outs

(Lagoudakis & Parr, ICML03)

Key idea

- determine the best action in each state
- train a classifier (e.g., decision tree) as a policy

API

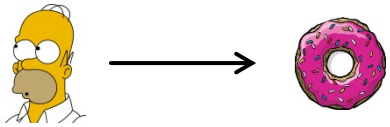
1. start with policy π_0
2. for each state s
   - evaluate all actions with roll-outs
   - determine the best action a^* (the one with highest est. Q-value)
   - generate a training example (s, a^*) if a^* is significantly better than all other actions in state s
3. use all training examples to train a policy $\pi: S \rightarrow A$
4. goto the 2nd step until stop

The policy is a classifier.

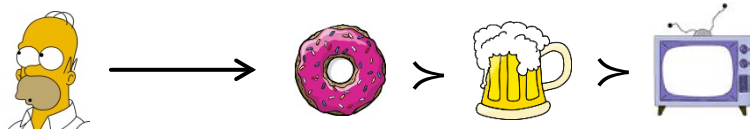
Label Ranking

The task in label ranking is to order a set of labels.

- Classification:
  predict one from a set of items



- Label ranking:
  predict a (partial or total) order realtion $\Pi(A)$ on a set of items A



Label rankers can be trained with label preferences

- we want to order actions based on the state decription
- rankers are trained on action preferences of the type $(s, a_i \succ a_j)$

Preference-Based Policy Iteration

Key idea

- determine preferences between pairs of actions
- train a label ranker as a policy

PBPI

1. start with policy π_0
2. for each state s
   - evaluate all actions with roll-outs
   - find action pairs (a_i, a_j) that a_i is significantly better than a_j
   - generate a training example $(s, a_i \succ a_j)$
3. use all training examples to train a policy $\pi: S \to \Pi(A)$
4. goto the 2nd step until stop

The policy is a label ranker.

Advantages of A Preference-Based Framework

Often, there is no natural numerical signal

– a preference-based formulation can deal with qualitative feedback

It is difficult to optimize multiple objectives

– a preference-based framework can flexibly define preferences over states according to multiple criteria (e.g., Pareto dominance)

It may impossible to determine the best action

– it is often easier to compare two actions

– in the case of roll-outs:

no training example for API

$a_3 \succ a_2$	6 of 10	*insignificant*
$a_3 \succ a_1$	10 of 10	*significant*
$a_2 \succ a_1$	9 of 10	*significant*

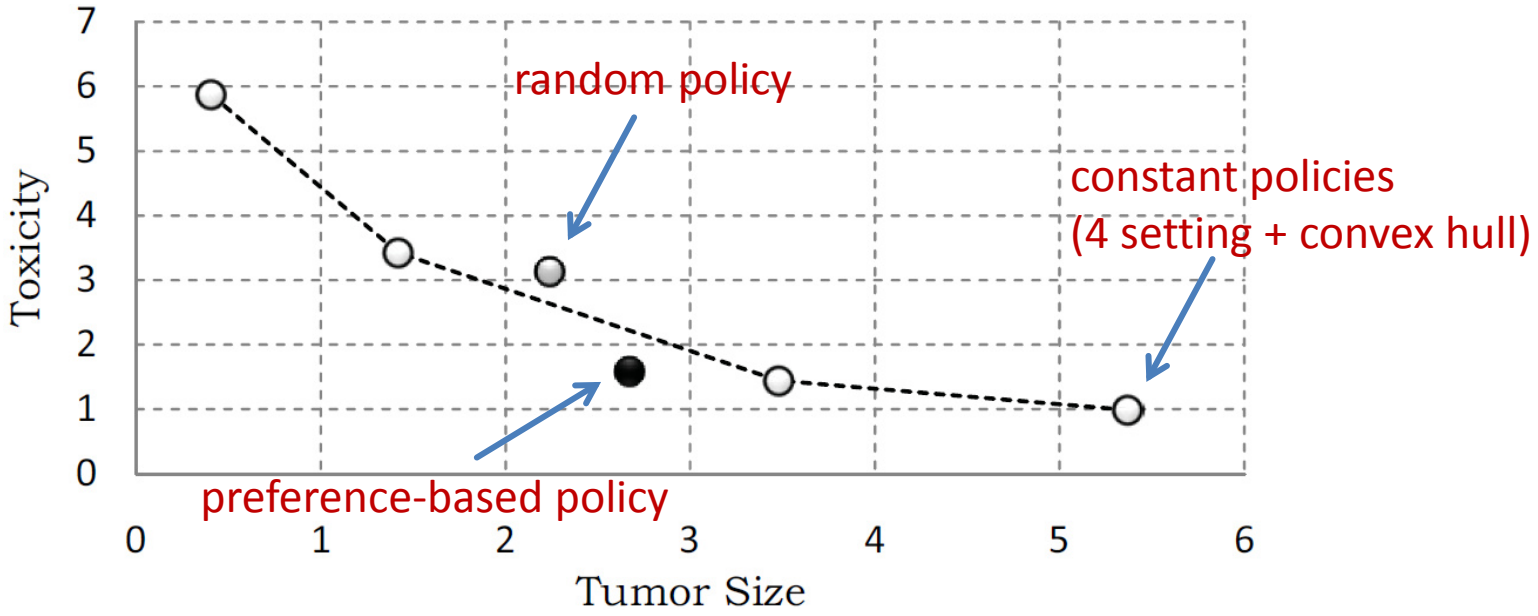
two training examples for PBPI

Case Study: Learning from Qualitative Feedback

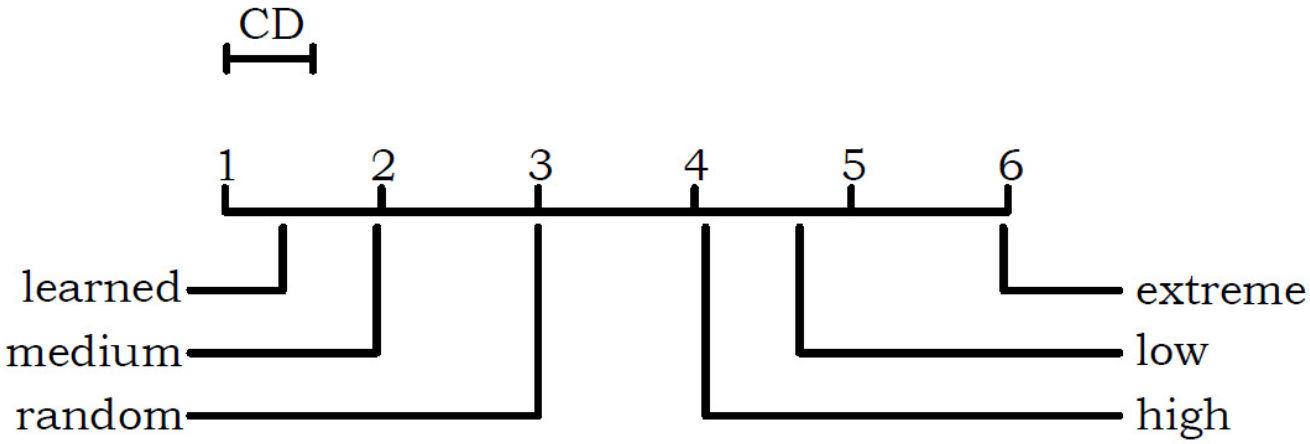
Empirical studies on the model of cancer clinical trials:

- 6-month treatment
- monthly chemotherapy with 4 dosage level
- action preferences generated via partial order relation with roll-outs
- 1000 patients for training, another 200 for testing
- the policy iteration stops when
  (1) the change of policies is smaller than a threshold, or
  (2) the number of policy iterations reaches 10

Case Study: Learning from Qualitative Feedback



Case Study: Learning from Qualitative Feedback



ranking w.r.t. the probabilities of surviving
during the whole treatment

Summary, Current and Future Work

- preference-based RL allows learning in a qualitative setting
- we proposed a preference-based extension of approximate policy iteration
- a case study on the cancer treatment problem

- ❖ theoretical foundation for preference-based RL
- ❖ What if we don't have a generative model? Is there an online version of preference-based RL?
- ❖ integration (qualitative) preference information and (quantitative) reward signals