Payam, Goodarzi
Fachbereich Mathematik und Informatik

Freie Universität Berlin

Intelligente Lehr-/Lernsysteme

Machine Learning in Education

Payam Goodarzi

26.02.2018

Overview

Prüfungsordnung: SPO von 2014 – Master Informatik			
Modul: Wissenschaftliches Arbeiten Praktische Informatik A			
Lehr- und Lernformen	**Präsenzstudium** (Semesterwochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)
Hauptseminar	2	Vortrag, schriftliche Ausarbeitung, regelmäßige Diskussionsbeiträge	Präsenzzeit HS: 30 Vor- und Nachbereitung HS: 60 Prüfungsvorbereitung und Prüfung: 60
Modulprüfung:	Schriftliche Ausarbeitung (ca. **4 500 Wörter**) mit mündlicher Präsentation (ca. **45 Minuten**); die Modulprüfung wird nicht differenziert bewertet.		
Arbeitszeitaufwand insgesamt:	**150 Stunden**	**5 LP**	

Overview

- Machine Learning
- Reinforcement Learning (RL)
- Markov Decision Process (MDP)
- Value Function
- Policy Iteration (Bellman Equation)
- Application of RL to induce pedagogical policies
- Summary
- References

Machine Learning

- **Supervised learning:**

  Given the input and labelled output, find the function, which maps the input to output.

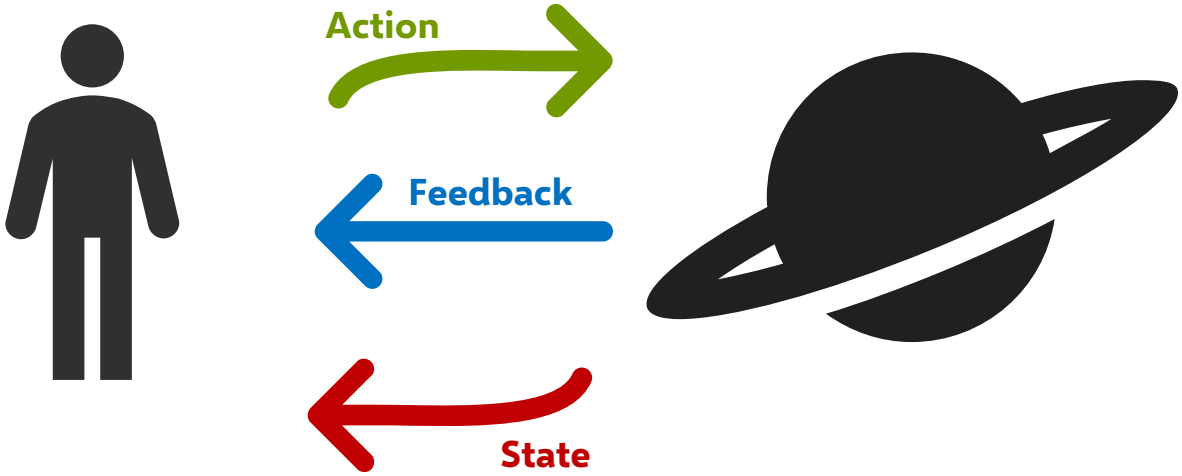
- **Unsupervised learning:**

  Given unlabelled data, infer the function, which describes the hidden structure in data.

- **Reinforcement learning:**

  Taking actions base on received rewards in an environment (Trial-and-Error)

Reinforcement learning (RL)



Reinforcement learning (RL)

By **model** we mean anything that an agent can use to predict how the environment will respond to its actions.

Planning refers to any computational process that takes a model as input and produces or improves a policy for interacting with the modeled environment.

- **Model-based RL:** Policy Iteration, Value Iteration, etc.
- **Model-free RL:** Q-Learning, Temporal difference learning, Monte Carlo, etc.

Markov Decision Process

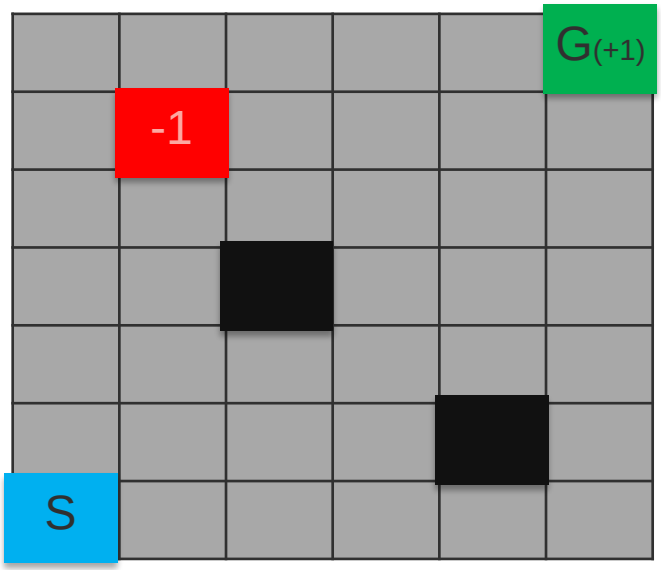
A RL task that satisfies the **Markov Property** is called a Markov Decision Process.

$$\Pr\{R_{t+1} = r, S_{t+1} = s' | S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\}$$

Environment's response at *t+1* depends only on the state and action representation at t

$$\Pr\{R_{t+1} = r, S_{t+1} = s' | S_t, A_t\}$$

States: S
Model: T(s, a, s') ~ Pr(s'| s, a)
Actions: A(s), A
Reward: R(s), R(s, a), R(s, a, s')
Policy: π(s) $\rightarrow$ a
Optimal policy: $\pi*$



Value Function

It estimates **how good** is for the agent to be in a given state.

- State-value function for policy π.

$$v^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, S_0 = s\right] \neq R(s)$$

- Action-value function for policy π.

$$q^{\pi}(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, S_0 = s, A_0 = a\right]$$

Bellman Equation

Value functions satisfy particular **recursive relationships** between value of a state and its possible successor states.

Bellman Equation averages over all the possibilities, weighting each by its probability of occurring.

$$v^{\pi}(s) = \sum_{a} \pi(a|s) \sum_{s'} p(s'|s,a) \left[r(s,a,s') + \gamma v_{\pi}(s') \right]$$

Optimal Policy

Optimal Policy is the policy, which maximizes the long-term expected reward (the greatest expected return).

$$\pi \geq \pi' \; if \; and \; only \; if \; v^{\pi}(s) \geq v^{\pi'}(s)$$

Optimal state-value function:

$$v^*(s) = \max_{\pi} v^{\pi}(s) \;\; for \; all \; s \; \epsilon \; S$$

Optimal action-value function:

$$q^*(s,a) = \max_{\pi} q^{\pi}(s,a) \;\; for \; all \; s \; \epsilon \; S \; and \; a \; \epsilon \; A(s)$$

Bellman Optimality Equation

Bellman Optimality Equation expresses the fact that the value of a state under an optimal policy must equal the expected return for the best action for that action.

Bellman Optimality Equation for v^*

$$v^*(s) = \max_{a \epsilon A(s)} q^{\pi^*}(s,a)$$

$$= \max_{a \epsilon A(s)} \sum_{s'} p(s'|s,a)\,[r(s,a,s') + \gamma v_*(s')]$$

Bellman Optimality Equation for q^*

$$q^*(s,a) = \sum_{s'} p(s'|s,a)\left[r(s,a,s') + \gamma \max_{a'} q_*(s',a')\right]$$

Policy Iteration

We can obtain a sequence of monotonically improving policies and value functions.

$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$$

E: evaluation, I: improvement

Each policy is guaranteed to be a strict improvement over the previous one.

Pseudocode of Policy Iteration:

- Initialization

$v(s) \in \mathbb{R}$ and $\pi(s) \in A(s)$ arbitrarily for all $s \in S$

Repeat

$\pi \leftarrow \pi'$

- Compute the value-function using π by solving

$$v(s) \leftarrow \sum_{s'} p(s'|s, \pi(s)) \left[r(s, \pi(s), s') + \gamma v(s') \right]$$

- Improve the policy at each state

$$\pi(s) \leftarrow \operatorname*{argmax}_a \sum_{s'} p(s'|s, a) \left[r(s, a, s') + \gamma v(s') \right]$$

Until $\pi = \pi'$

Applying RL to induce pedagogical policies

- The behaviors of e-learning system can be considered as a **sequential decision process**.
- **Pedagogical strategies** are the **policies**.
- We investigate the project of a team consisting of people of Carnegie Mellon University, Arizona State University and University of Pittsburgh USA.
- They applied **Policy Iteration** to improve the effectiveness of an Intelligent Tutoring system (ITS) by inducing pedagogical policies directly from an **exploratory corpus**.

- The project has to phases:
  1. Training phase:
     - Defining state representation S, action space A, reward function R.
     - Collecting a training corpus Γ.
  2. Test phase: Evaluation of RL-induced policies.

Exploratory Corpus

- Collecting the exploratory corpus by training **human student** on ITS that makes random decisions.
- The ITS is **Cordillera** a natural language (NL) tutoring system which teaches students introductory physics.

Actions Set

- Two type of decisions:
  - **Elicit/Tell** (ET): elicit the next step from Student or tell a student the next step directly.
  - **Justify/Skip-justify** (JS): ask students to justify a step they have taken.

- The ITS is **Cordillera** a natural language (NL) tutoring system which teaches students introductory physics.

Actions Set – ET/JS

(a) Justify Version

1. **T:**Can we infer the direction of the velocity of the rock at T1 from the rock's kinetic energy at T1? **{ELICIT}**
2. **S:**nope.
3. **T:**Excellent! Please explain why. **{JUSTIFY, ELICIT}**
4. **S:**Only the magnitude of the velocity and not the direction of it is part of the definition of kinetic energy.
5. **T:**Excellent! Now that we know v1, $\cdots$.

(b) Skip-justify Version

1. **T:**Can we infer the direction of the velocity of the rock at T1 from the rock's kinetic energy at T1? **{ELICIT}**
2. **S:**nope.
3. **T:**Excellent! **{Skip-JUSTIFY}**
4. **T:**Now that we know v1, $\cdots$.

(a) Elicit Version

1. **T:** So let's start with determining the value of v1.
2. **T:** Which principle will help you calculate the rock's instantaneous magnitude of velocity at T1? **{ELICIT}**
3. **S:** definition of kinetic energy
4. **T:** Please write the equation for how the definition of kinetic energy applies to this problem at T1 **{ELICIT}**
5. **S:** ke1 =0.5*m*v1^2
6. **T:** From KE1 = 0.5*m*v1^2, $\cdots$

(b) Tell Version

1. **T:** So let's start with determining the value of v1.
2. **T:** To calculate the rocks instantaneous magnitude of velocity at T1, we will apply the definition of kinetic energy again. **{TELL}**
3. **T:** Let me just write the equation for you: KE1 = 0.5*m*v1^2. **{TELL}**
4. **T:** From KE1 = 0.5*m*v1^2, $\cdots$

State Representation (State features)

- An effective state representation *S* should be accurate and compact model of the learning context.
- From **MDP** perspective, pedagogical strategies are simply a set of policies.
- A state in the MDP might be a set of **features**.
- Identifying useful **state features** is challenging.

- A series of **feature selection** procedure had been used:
  - Four RL-based feature selection methods
  - PCA-based feature selection method
  - Four PCA and RL-based feature selection methods
  - Random feature selection methods

Reward function and Evaluation Metrics

- **Expected cumulative reward:**

$$ECR_\pi = \sum_{i=1}^{n} \frac{N_i}{N_1 + \cdots + N_n} * V(S_i)$$

N_i: number of time that S_i appears, $V(S_i)$: value of state S_i

The higher ECR, the better the policy is supposed to perform.

- **Confidence Interval (CI):** 95% CI [Lower_Bound, Upper_Bound]
- **Hedge:**

$$Hedge = \frac{ECR}{Upper\_Bound\ - Lower\_Bound}$$

- **Reward function:**

Normalized learning gain (NLG)

$$NLG = \frac{posttest\ - pretest}{1 - pretest}$$

Features

- **Autonomy (A)**: amount of work performed by student.
  Exp: [tellsSinceElicitA]
- **Background (BG)**: general background information about student.
  Exp: [gender, age, MathSAT,...]
- **Problem Solving Contextual (PS)**: Exp: [StepSimplicityPS]
- **Performance (PM)**: Exp: [pctCorrectKCPM]
- **Student dialogue**: characterizes student language.
  Exp: [stuAverageWordSD]
- **Temporal Situation (T)**: time related information about problem solving
  Exp: [durationKCBetweenDecisionT]

Example

An example of selected „best“ policy on ET decisions

$[\mathbf{S:}] = \{StepSimplicityPS \times TuConceptsToWordsPS \times TuAvgWordsSesPS\}$

$[\mathbf{A:}] = \{Elicit, Tell\}$

[Policy:]

rules 1-5: $\begin{bmatrix} 0:0:0 \\ 0:0:1 \\ 1:0:1 \\ 1:1:0 \\ 1:1:1 \end{bmatrix}$ → **Elicit**

rule 6: $\begin{bmatrix} 0:1:0 \end{bmatrix}$ → **Tell**

rules 7-8: $\begin{bmatrix} 0:1:1 \\ 1:0:0 \end{bmatrix}$ → **Either Elicit or Tell**

ECR: 14.25

95%CI: [10.04, 18.12]

Procedure

All participants in this project experienced the same five standard phases:

1. Background survey
2. Pre-training
3. Pre-test
4. Training
5. Post-test

		Defined features	Feature occurrences
1	Autonomy (A)	5	8
2	Background (BG)	5	1
3	Performance (PM)	12	5
4	Problem Solving Contextual (PS)	15	30
5	Student Dialogue (SD)	10	8
6	Temporal Situation (T)	3	7
7	Total	50	59

Occurrence of six category features in the final NormGain tutorial policies.

Summary

1. Choose an **appropriate Reward measure**, an **appropriate list of Features** for the state representations, and identify a set of reasonable **system Decisions.**

2. Build an initial training system that collects an **exploratory dataset**.

3. Apply **feature selection** methods, to select a subset of features that capture the most effective factors in the learning environment. Then use the exploratory corpus to build an empirical **MDP model** for the subset of state features.

4. Compute the **optimal dialogue policy** (by **Policy Iteration**) according to this learned MDP.

5. Add the learned policy to the system and **evaluate** the policy on a new group of users.

References

- Chi, Min, et al. "Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies." *User Modeling and User-Adapted Interaction* 21.1-2 (2011): 137-180.

- Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. Vol. 1. No. 1. Cambridge: MIT press, 1998.

- Reinforcement Learning lectures by David Silver from DeepMind http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html

- Szepesvari, Csaba. "Algorithms for Reinforcement Learning (Synthesis Lectures on Artificial Intelligence and Machine Learning)." *Morgan and Claypool* (2010).

- Ai, Hua, and Diane J. Litman. "Knowledge consistent user simulations for dialog systems." *Eighth Annual Conference of the International Speech Communication Association*. 2007.

Thanks for your Attention!