Reinforcement Learning Interview — Csaba Szepesvari — 2016.11.24

1. **Maybe starting at how you joins RLAI group.**
    a. My learning all of it was a longer story. I got into reinforcement learning during my PhD so try to escape neural network, (no you can't write this stuff.) OK OK, but this is true. but I'm sure when I was doing my Masters my first encounter was AI and neural network. And I found I'm on the top of the cool and I had all the books that I could have access to try to research it and try to prove theorems about it because my degree is in pure mathematics so I'm interested in figuring out what you can do it all, rather than just trying to sing songs maybe one day they've heard next day. If you prove that it works then they did work, right? And so I had this and this drive and looking at neural network but couldn't figure them out. I was working on (done for) , I like trying to figure like I put up some negative besides if you can do this you can do that, you know, at the time this has become fashion but then I found reinforcement learning and it seems that people who are able to say positive things about the reinforcement learning and it has (cool map? 1:34) and also I was primarily driven by my interest in AI and figuring out what the mind does. It's really like high school, I really wanted to figure out the how people think, I was even thinking about getting into Biology, and then my friends told me that no you have to study a lot. I'm not going to do that. Lastly sure was interested in mind, no Biology but Reinforcement learning seems to be a topic that is closest tool to study like how to come to decisions with long term of sequential studies, figuring out down certain environments. It's all in one package, it's beautiful Math. So I was totally hang on it. Once I saw it, OK, I'm in, so I have to do this. So that my PhD was about reinforcement learning, I was researching these things.
    b. And how I got here was, I knew Richard Sutton. We met at conference, under circumstances that I can't describe but it was fun. Then he told me that there(UAlberta) is an opportunity that I could come here. At first I didn't believe that he was not kidding, then I come here to visit this lovely place and lovely people. So it's how it happened.
2. **What makes you decide to stay here(UAlberta)?**
    a. Skiing was a big part of it. Oh so the people who are like amazingly kind when I got here. It was like something special I never had before. And also like it's a very good researching mind. So it's like a whole package was really nice and we didn't mind cold.
3. **You have done a lot of works and implemented a lot of algorithms and you're also very famous on online learning. Could you explain in a general way that maybe allow our audience understand for all your works in RLAI.**
    a. There's a situation that you have an interest but you don't have much time. People are inclined to, you know, make a little understand learning. I mean it's a short book that has short purpose. Because people have not much time to read but the book is still of a lot of information. It's called the <Algorithms of Reinforcement Learning>.
    b. UCT (*Upper Confidence Bound applied to trees*), the first Monte Carlo search algorithms, which was better theoretically founded than the others, and then to be pretty success

and was used in many application domain. This is an algorithms for planning, answers to environment and people choose it then use it in many different settings. So maybe that's one of my important contributions.

c. Bandit Problem("partial monitoring problem"). I'll describe it through an example. So let's say you want to teach kids to learn math, and what you can do is that you can try all kinds of different methods, which are limited in a finite set. Every methods to some kind of students will give you a feedback or an observation. And the key point is that those observation may be not valid, which is, students' reaction cannot directly tell you about whether they get into it or not, there are always noises between you and students. If you're a good teacher, you can definitely acquire some useful information through the interaction between you and your students. For computer programs, it will try different things and grab results of different observations coming out, and you can make sure that things go into right direction. This is very general. So the difference of Reinforcement Learning is that RL has its underlying states. Let's say that you have finished your teaching task for one student, which means you got some observation. And then the next student comes in, you know that you may need take different methods or actions, and you start from scratch. It's kind of like a sequence of students and they are from different background, So the important thing is that your knowledge with previous students cannot directly be transferred. This means you have to try again and again, through the process of interacting with students and observing the result, you start to learn a useful pattern and you get to know what works and doesn't work. After repeating for a long time, we're going to get better over time. So why this is called "partial monitoring" is because you don't have full information about how it works. As you are taking different actions, you gained some knowledge of past students. At this point, you may be disappointed because your knowledge may not be applied to current students while you expect some good feedback or observation, they even don't respond to you. If you still repeat your past experience, then students get tired and nothing works. So you have to think about whether your action is better to take to acquire more information or is it better to take an action to get best rewards from the experience before.

d. **If we don't keep state in the implementing algorithms and how we actually could know about them?**

    i. The difference is that the algorithms can keep states, but the environment doesn't keep them. OK So there you are going through a sequence of situations and the each situation is similar to the previous one and there is no transfer between them. It seems that algorithms immediately forget the environment. It is like that all customers come to your website, and they're talking to each others. Whatever you offer to a customer, it doesn't matter for the other. If they would start talking then that would change things but many times you can just assured all of this is just happening. Like one student is similar to the other student but if you're teaching one and you have some interaction with him, it doesn't mean the others' minds are going to change. Like students and customers, there is no state. But in another example, if customer has a longer session, of course there are states. So it's a simplification to say that there is no state. But you start at simpler setting and you will make stronger assumption like that there is no state. Once you understand simpler setting, you can go on and try to understand more complicated setting.

4. **Why do you escape Neural Network and How do you think of Deep Learning?**

a. NN may not be proved in mathematics ways, I always bumped into some negative sides and I was struggling. I'm still struggling to understand how the NN ever works. And I think of a lot of other people also keep asking the same questions: why is it by fact? If given a very clean situation and there is no algorithms that is able to do in polynomial time, so it shouldn't work. But it works! But at a time I discovered RL and I tracked to it because I could put it in mathematical way.

b. DL. I'm totally intrigued by DL. And it's very cool. Last semester we had a class, my students who were learning on the theory of the papers and trying to understand fine how the DL works. Well there is a lot to improve once you're understanding of why it works. But at first it is very limited, maybe there is a very simple tool to understand these things better. In certain settings, local minimum is not bad. But we should understand a little bit better because those designs are usually for a very particular purpose so you have a clear clean Mathematica for combinations and simplified ones. When we have a similar situation, it must be working but why it is working? Does it happen ? We don't quite know. But these are very interesting and intriguing question. I think all theories must be proved, once we got them proved we can understand. But others may say you can program it then you can understand it. That's also true. If you have theory to prove that on the assumptions you show that something work and something doesn't work. This is very strong statement so it's very desirable to have those it's not necessarily easy to get those, you need to have some guesses then you're starting to get something like this. So we already know that we know actual assumptions. Nothing new about why the NN works. But what are the set of assumptions under which they do work.

5. **What's your feeling about Deep Reinforcement Learning?**

   a. It is my favorite subject but it's very interesting that it's someone have to do it and that it is good that they have done it. And all the ideas in RL are estimated by a function, improving policies. All those ideas are made to work if you have right computational resources and flexible enough approximation techniques like NNs. If you have flexible enough architectures and these are good enough to work. So there is theoretical explanation of why the algorithms, by the way, the same algorithms, if you try them with not so flexible architecture, they tend not to work. A lot of people have experienced that before that they try a small NN, so the sort of other type of functional approximation techniques are the same as used and model of the same problem. NNs tend not to give a good result.

6. **[26:00~27:00]**问题有点长没搞懂问的啥

   a. Could think about machine learning, is that theory and practice that kind of go hand in hand. Practitioners like I hold that there are sometimes this thing to practition definitely. Looking at what's happening in practice and much of the theory people or the is strongly motivated by like what are the needs of our practitioners and so when I see it is practice enhancement but I think that 'hey, that's a good problem for me to look at. why did this work, I need to understand.' So that is my viewpoint. It's great that you've heard that it's that one example is going to work tomorrow on a different example(意思应该是在不知道具体如何工作的情况下某个例子今天可以成立, 按照同样的方法换另一个例子在明天也能用; 像是用英文训练成功的NN 换做用中文训练也能工作), if we can build sound theory. And we'll be in a good position to maybe predict on what data you should be running these architectures or those ar-

chitectures. This is how you should be changing your algorithms to get good results. it's pretty interesting that these are happening and I think it's a great time because there are so many scenes that you don't understand. So it's a great time to be a practitioner for many many openquestion.

7. **About training NNs.**

    a. You can pick one up for any stochastic example but it doesn't really matter what you pick and you can tweak it a little, such that for a human, the two images, untweak one and tweak one, look exactly the same and you're tweaking the image such a way the Neural Network is going to be Complete Wrong on the tweak image where it was complete sure about what's on the image. It may be a cat and it will always be a cat, but NN may recognize it as a dog when there are only a little changes in the image. Those happenings show that something is not quite right though NN could successfully recognize the object but through this example we may realize that there's a chance that these NNs are crazily overfitting. What strange is that things like this are happening everywhere from the past to now. It seems to be a phenomenon. So this bothers me and bothers a lot of people. So people try to design better learning algorithms that safeguard against these overfitting phenomenon. I think that this is a very example to show that you can overfit huge test sets if you're doing a big gradient search on the space of NNs' architecture. So this overfitting seems to be a problem and so it would be nice to have more robust training methods.