# Model-free POMDP optimisation of tutoring systems with echo-state networks

**Lucie Daubigney**[1,3] **Matthieu Geist**[1] **Olivier Pietquin**[1,2]
[1]IMS-MaLIS – Supélec (Metz, France), [2]UMI2958 – GeorgiaTech/CNRS (Metz, France)
[3]Team project MaIA – Loria (Nancy, France)

## Abstract

Intelligent Tutoring Systems (ITSs) are now recognised as an interesting alternative for providing learning opportunities in various domains. The Reinforcement Learning (RL) approach has been shown reliable for finding efficient teaching strategies. However, similarly to other human-machine interaction systems such as spoken dialogue systems, ITSs suffer from a partial knowledge of the interlocutor's intentions. In the dialogue case, engineering work can infer a precise state of the user by taking into account the uncertainty provided by the spoken understanding language module. A model-free approach based on RL and Echo State Newtorks (ESNs), which retrieves similar information, is proposed here for tutoring.

## 1 Introduction

For the last decades, Intelligent Tutoring Systems (ITSs) have become powerful tools in various domains such as mathematics (Koedinger et al., 1997), physics (Vanlehn et al., 2005; Litman and Silliman, 2004; Graesser et al., 2005), computer sciences (Corbett et al., 1995), reading (Mostow and Aist, 2001), or foreign languages (Heift and Schulze, 2007; Amaral and Meurers, 2011). Their appeal relies on the fact that each student does not have to follow an average teaching strategy, especially as the one-to-one tutoring has been proven the most efficent (Bloom, 1968). The expertise of a teacher relies on his capacity to advice at the right time the student to acquire new skills. To do so, the teacher is able to choose iteratively pedagogical activities. From this perspective, teaching is a sequential decision-making problem. To solve it, the reinforcement learning (Sutton and Barto, 1998) approach and the Markov Decision Process (MDP) paradigm have been successfully used (Iglesias et al., 2009). Given a situation, each teacher's decision is locally quantified by a *reward*. However, the consequences of the teacher's actions on the student's cognition cannot be exactly determined, which introduce uncertainty.

To find a solution, one can notice that spoken dialogue management and tutoring are closely related. Both are humain-computer interactions in which the human user's intentions are not perfectly known. In the spoken dialogue case, the partial observability is due to the recognition errors introduced by the speech understanding module. They are taken into account by using some hypotheses about how the language is constructed. Thus, accurate models to link observations from the user's recognised utterances to the underlying intentions can be set up. For example, the Hidden Information State paradigm (Young et al., 2006; Young et al., 2010) builds a state which is a summary of the dialogue history (Gašić et al., 2010; Daubigney et al., 2011; Daubigney et al., 2012). However, in the ITS case, such a state is harder to develop since the cognition cannot be determined by analysing a physical signal. Thus, a model-free approach is preferred here.

To do so, a memory of the past observations and actions is built by means of a Recurrent Neural Network (RNN) and more precisely an Echo State Network (ESN) (Jaeger, 2001). The internal state of the network can be shown (under some resonable conditions) to meet the Markov property (Szita et al., 2006). This internal state is then used with a standard RL algorithm to estimate the optimal solution. It has already been applied to RL in (Szita et al., 2006) in limited toy applications and it is, to our knowledge, the first attempt to use it in an interaction framework. The proof of concept presented in Szita's article uses the common SARSA algorithm which is an *on-line* and *on-policy* algorithm. Each improvement of the strat-



*Proceedings of the SIGDIAL 2013 Conference*, pages 102–106,
Metz, France, 22-24 August 2013. 

egy is directly tested. In the case of teaching, testing poor decisions can be problematic. Here, we thus propose the combination of an ESN with an *off-line* and *off-policy* algorithm, namely the Least Square Policy Algorithm (LSPI) (Lagoudakis and Parr, 2003), which is another original contribution of this paper. Indeed, learning the solution with Partially Observable MDPs in a batch and off-policy manner is not common in the literature.

## 2 Markov Decision Process and Reinforcement Learning

Formally, an MDP is a tuple $\{S, A, T, R, \gamma\}$ set up to describe the tutor environment. The set $S$ is the *state space* which represents the information about the student, $A$ is the *action space* which contains the tutor's actions, $T$ is a set of *transition probabilities* defined such that $T = \{p(s'|s,a), \forall(s',s,a) \in S \times S \times A\}$, $R$ is the *reward function*, given according to the student progression for example, and $\gamma \in [0,1]$ is the *discount factor* which weights the future rewards. The set of transitions probabilities in the ITS case is unknown: the evolution of the student intentions cannot be determined. Solving the MPD consists in finding the optimal strategy, called the optimal policy which brings the highest expected cumulative reward.

However, in the ITS case, information about the student's knowledge, represented by $s$, can only be known through observations. Let $O = \{o_i\}$ be the set of possible observations. Yet, if only observations are available, a memory of what happened during previous interactions (the history) is necessary, because the process of observations does not meet the Markov property. The history is the sequence of observation-action pairs encountered during a whole teaching phase. Let $H = \{h_i\}$ be the set of all possible histories with $h_i = \{o_0, a_0, o_1, a_1, ..., o_{i-1}, a_{i-1}, o_i\}$.

When the POMDP framework is used, the underlying state $s_i$ is inferred from the history by means of a model of probabilities linking $s_i$ to $h_i$. In the case of human-machine interactions, this model is not available. It can be approximated but the considered solutions are *ad-hoc* to a particular problem, thus difficult to reuse. Here, we propose an approach with as few assumptions as possible about the student cognitive model by using Echo States Networks (ESNs). This approach builds a compact representation of the history space $H$.

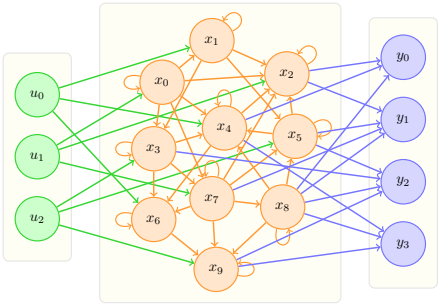


Figure 1: RNN structure (for sake of readability, all the connections do not appear).

## 3 Echo State Networks

An Echo State Network is represented by three layers of neurons (Fig. 1): an input, a hidden and an output. The number of neurons in the hidden layer is supposed to be large and each of them can be connected to itself. These recurrent connections are responsible for reusing the value of the neurons at a previous time step. Consequently, a memory is built in the reservoir and trajectories can be encoded. Only the connections from the hidden layer to the output one are learnt since all the other connections are randomly and sparsely set. The recurrent connections are defined so that the echo state property is met (Jaeger, 2001): if after a given number of updates of the input neurons, two internal states are exactly the same, then the input sequences which led to these two internal states are identical.

The connections of the ESN are presented in Fig. 2, with $u_k \in \mathbb{R}^{N_i}$, $x_k \in \mathbb{R}^{N_h}$ and $y_k \in \mathbb{R}^{N_o}$, respectively representing the values of the input, hidden and output layers, $N_i, N_h$ and $N_o$ being the respective number of neurons and $W^{in} \in \mathbb{M}_{N_h \times N_i}$, $W^{hid} \in \mathbb{M}_{N_h \times N_h}$ and $W^{out} \in \mathbb{M}_{N_o \times N_h}$, matrix containing the synaptic weights. After a training, the output $y_k$ returns a linear approximation of the internal state of the reservoir. This output depends on the sequence of inputs $u_0, \cdots, u_k$ and not only $u_k$, through $x_k$.

Combining ESNs and RL is of interest. By means of the echo state property, a summary of the observations and decisions encountered during the tutoring phase is provided through the internal state $x$. In (Szita et al., 2006), it has been proven to meet the Markov property with high probability. It thus can be used as a state for standard RL algorithms. Here, more precisely, it represents the basis function of an approximation of the Q-

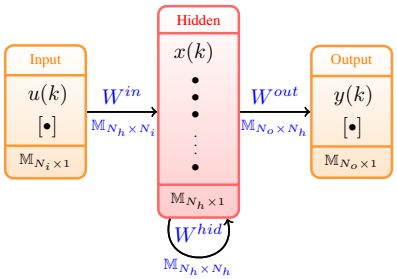


Figure 2: Structure of an ESN. For the example, $N_i = 1$ and $N_o = 1$.

function. This function is associated with a policy $\pi$, defined for each couple $(s, a) \in S \times A$ such that $Q^\pi(s, a) = E\left[\sum_i \gamma^i r_i | s_0 = s, a_0 = a\right]$ and quantifies the policy. ESNs are used in the following way to solve RL problems. The network is responsible for giving, from an observations $o_k$ and an action $a_k$ at time step $k$, a linear estimation of the value of the Q-function $\hat{Q}_\theta(h_k, a_k)$ (with $h_k = \{o_0, a_0, ..., o_{k-1}, a_{k-1}, o_k\}$). The state $s$ is not used in the estimation of the Q-function since it is unknown. Instead, it is replaced by the history $h_k$. The input of the ESN, $u_k$, is thus the concatenation of the observation $o_k$ and the action $a_k$: $u_k = (o_k, a_k)$. The internal state $x_k$ which component are in $[-1, 1]$, is a summary of the history $h_k$ and the action $a_k$. Thus, the estimation of the Q-function is $\hat{Q}_\theta(h_k, a_k) = \theta^\top x_k$. The values of the output connections are learnt by means of the LSPI algorithm. With this algorithm, the optimal policy is learnt from a fixed set of data.

## 4 Experimental settings

For the experiments, we assume that the teaching can be done by means of three actions. First, a lesson can be presented to make the knowledge of the student increase. The second and third actions are evaluations. They can either be a simple question or a final exam. The final exam consists in asking a hundred yes/no questions of equal complexity and on the same topic. The student does not have a feedback. Once it is proposed, a new teaching episode starts. Three observations are returned to the ITS. If a lesson is proposed to the user, the observation is neutral: no feedback comes from the student since the direct influence of the lesson remains unknown. The two other obervations appear when a question is asked (yes or no). Consequently, one observation is not enough to choose the next action since no clue is given about how many lessons have led to this result. A non-null reward is only given when a final exam is proposed. In this case, it is proportional to the rate of correct answers among all the answers given during the exam. Thus, each improvement is taken into account. The $\gamma$ factor is set to $0.97$.

In this proof of concept, the results have been obtained with simulated students from (Chang et al., 2006) to ensure the reproducibility of the experiments. The simulation implements two abilities: answering a question and learning with a lesson. Three groups of students have been set up. The first one, $T1$, is supposed to be able to learn very efficiently, the second, $T2$, needs a few more lessons to provide good answers, and the third, $T3$, needs a lot of lessons to answer correctly.

## 5 Results

Several teaching strategies have been compared. As a lower bound baseline, a random strategy has been tested. With a probability (w.p.) of $0.6$, a lesson is proposed, w.p. of $0.2$ a question is chosen, and w.p. of $0.2$ a final exam is proposed. The data generated with this random strategy have been used by the LSPI algorithm and an informed state space. The second baseline proposed is the reactive policy learnt by LSPI (called *reactive-LSPI*), only from obervations. Neither the information about the number of lessons proposed nor the internal state of the ESN is used. The third strategy is learnt by using the observations and a counter of lessons already given (called *informed-LSPI*). Thus, this state supposedly contains sufficient information to take the decision. For this case, since the numbers of observations and lessons are discrete thus countable, a tabular representation is chosen for the Q-function. The fourth strategy uses the internal state of the ESN as basis function for the Q-function (called *ESN-LSPI*). There are $50$ hidden neurons. Different sizes of training data sets are tested. Among the data, the three types of students are represented in equal proportions. One hundred policies are learnt for each of the methods presented, except for the ESN-LSPI. For this one, $10$ ESNs are generated and $10$ training sessions are performed with each one of them. The mean over the average results of each of the $10$ learnings is presented in the results. Each of the policies have been tested $1000$ times.

Fig. 3 shows a comparison of the learnt strategies. The three types of students are used for the training and test phases. One can notice that

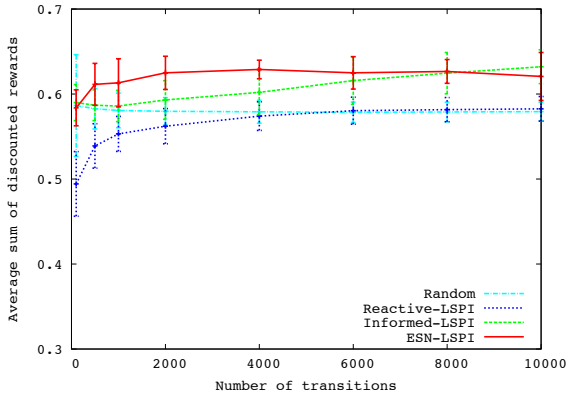


Figure 3: Comparison of the different strategies.

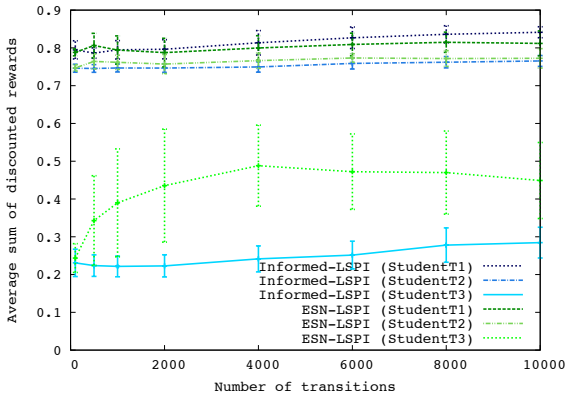


Figure 4: Results of the learnt policies for each group of students.

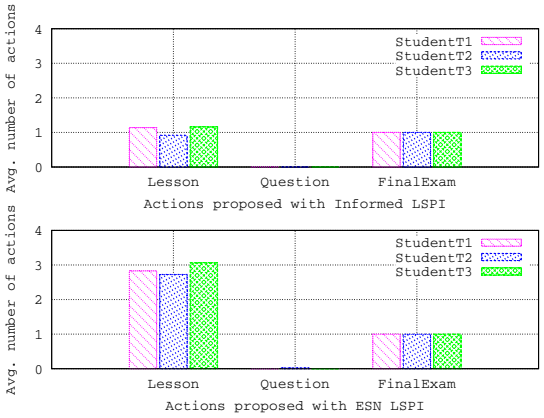


Figure 5: Distribution of the actions (the size of the training dataset is 10000).

the standard deviation is larger when the ESN are used because uncertainty is added when generating the ESN since the connections are randomly set. The random and the reactive policies give the poorest results. Yet, the average reward increases because of the data in the training set. For small sets, long sequences of lessons only have not been encountered. Thus, larger rewards have not been encountered either. For the two other curves, with a reasonable number of interactions (around 8000), a good strategy is learnt by using informed-LSPI. The strategies learnt with the ESN require fewer transitions and allow a faster learning. In this case, the optimum is reached with 2000 transitions while 8000 ones are needed to reach the same quality with the informed-LSPI strategy. Around 10000 samples, both policies give the same results. However, less information is given in the ESN approach (only observations). Thus, this approach is more generic. The counter information may not be sufficient for more complex problems.

To compare the efficiency of the learnt policies, the informed-LSPI and ESN-LSPI are plotted for each group of students in Fig. 4. All the strategies are learnt with the same data sets than previously, but only one type of students is tested at a time. For the $T2$ and $T3$ types, the average results are better with ESN-LSPI (especially for the $T3$ type). For the $T1$ group, informed-LSPI returns slighlty better results. A better insight of the behaviour of each policy is given in Fig. 5 by plotting the distribution of the actions used during the test phase. A comparison reveals that the number of lessons is higher in the ESN-LSPI case (around 3) whereas only one lesson is given in average with informed-LSPI. This is of benefit to students of the third group and thus implicitly to those of the first and second groups. The number of lessons is even larger for the third group than for the two others (0.5 more in average). However, in the informed-LSPI case, the learnt policy is only profitable for those of the first group, who are already skilled (this conclusion is consistent with the Fig. 4). Questions are very rarely asked because once the number of lessons has been learnt, they bring no more information.

## 6 Conclusion

We proposed a model-free approach which uses only observations to find optimal teaching stategies. A summary of the history encountered is implemented by means of an ESN. This summary has been proven to be Markovian by (Szita et al., 2006). A standard RL algorithm which can learn from already collected data, is then used to perform the learning. Preliminary experiments have been presented on simulated data. In future works, we plan to apply this method to SDSs.

**Acknowledgments**
Results have been computed with the InterCell cluster funded by the Région Lorraine.

## References


L. Amaral and D. Meurers. 2011. On using intelligent computer-assisted language learning in real-life foreign language teaching and learning. *ReCALL*, 23(1):4–24.

B. Bloom. 1968. Learning for mastery. *Evaluation comment*, 1(2):1–5.

K. Chang, J. Beck, J. Mostow, and A. Corbett. 2006. A bayes net toolkit for student modeling in intelligent tutoring systems. In *Intelligent Tutoring Systems*, pages 104–113. Springer.

A. Corbett, J. Anderson, and A. OBrien. 1995. Student modeling in the act programming tutor. *Cognitively diagnostic assessment*, pages 19–41.

L. Daubigney, M. Gašić, S. Chandramohan, M. Geist, O. Pietquin, and S. Young. 2011. Uncertainty management for on-line optimisation of a POMDP-based large-scale spoken dialogue system. In *Proceedings of Interspeech'11*.

L. Daubigney, M. Geist, S. Chandramohan, and O. Pietquin. 2012. A Comprehensive Reinforcement Learning Framework for Dialogue Management Optimisation. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):891–902.

M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young. 2010. Gaussian processes for fast policy optimisation of POMDP-based dialogue managers. In *Proceedings of SIGdial'10*.

A. Graesser, P. Chipman, B. Haynes, and A. Olney. 2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on*, 48(4):612–618.

T. Heift and M. Schulze. 2007. *Errors and intelligence in computer-assisted language learning: Parsers and pedagogues*, volume 2. Psychology Press.

Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. 2009. Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. *Applied Intelligence*, 31(1):89–106.

H. Jaeger. 2001. The "echo state" approach to analysing and training recurrent neural networks. Technical report, Technical Report GMD Report 148, German National Research Center for Information Technology.

K. Koedinger, J. Anderson, W. Hadley, M. Mark, et al. 1997. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED)*, 8:30–43.

M. Lagoudakis and R. Parr. 2003. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149.

D. Litman and S. Silliman. 2004. Itspoke: An intelligent tutoring spoken dialogue system. In *Demonstration Papers at HLT-NAACL 2004*, pages 5–8. Association for Computational Linguistics.

J. Mostow and G. Aist. 2001. Evaluating tutors that listen: an overview of project listen. In *Smart machines in education*, pages 169–234. MIT Press.

R. Sutton and A. Barto. 1998. *Reinforcement learning: An introduction*. The MIT press.

I. Szita, V. Gyenes, and A. Lőrincz. 2006. Reinforcement learning with echo state networks. *Artificial Neural Networks–ICANN 2006*, pages 830–839.

K. Vanlehn, C. Lynch, K. Schulze, J. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. 2005. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3):147–204.

S. Young, J. Schatzmann, B. Thomson, H. Ye, and K. Weilhammer. 2006. The HIS dialogue manager. In *Proceedings of IEEE/ACL Workshop on Spoken Language Technology (SLT'06)*.

S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.