What Are Some of the Best Ways for a Machine to Learn?

Amir Ramezani Dooraki, Ramezani Amir

HAL Id: hal-01822073
https://hal.science/hal-01822073

What Are Some of the Best Ways for a Machine to Learn?

Amir Ramezani Dooraki

Abstract Artificial intelligence recently had a great advancements caused by the emergence of new processing power and machine learning methods. Having said that, the learning capability of artificial intelligence is still at its infancy comparing to the learning capability of human and many animals. Many of the current artificial intelligence applications can only operate in a very orchestrated, specific environments with an extensive training set that exactly describes the conditions that will occur during execution time. Having that in mind, and considering the several existing machine learning methods this question rises that 'What are some of the best ways for a machine to learn?'
Regarding the learning methods of human, Confucius' point of view is that they are by experience, imitation and reflection. This paper tries to explore and discuss regarding these three ways of learning and their implementations in machines by having a look at how they happen in minds.

Keywords Artificial Intelligence · Supervised Learning · Reinforcement Learning · Unsupervised Learning · Machine Imagination · Machine Learning · Cognitive Development

1 Introduction

How minds work, or in another word how a human brain thinks, with the goal of implementing it in machines, is a long-term question in artificial intelligence. In the recent years, Artificial Intelligence algorithms have demonstrated outstanding progress and nowadays can be found in a variety of applications such as autonomous vehicles, computer games and health care automations. Having said that, the learning capability of artificial intelligence is still in its infancy comparing to the learning capability of humans. They can only operate in a very orchestrated, specific environments with an extensive training set that exactly describes the conditions that will occur during execution time. Systems of this kind are limited to the expertise and educated guesses of their human programmer; they lack the ability to learn in, or tune themselves to real-world environments autonomously or to employ learning in novel situations.

In order to create smarter machines, with an abstract similarity to human minds, design and development of complicated computational algorithms inspired by natural intelligence has already started. Among bio-inspired algorithms those which simulated cognitive architectures are crucial where they are "the computational implementation of a cognitive model, and as such, constitute the substrate for all the cognitive functionalities in robots, like perception, attention, action selection, learning, reasoning, etc" [1].
As a result of these efforts, there are several successful methods of machine learning such as reinforcement learning, supervised learning and unsupervised learning where each group has its own branches: temporal difference and actor-critic methods for reinforcement learning, artificial neural network and its deep architectures for supervised learning and self-organizing and clustering methods for unsupervised learning. Saying that, this question rises that "What are some of the best ways for a machine to learn?"
In this paper, I try to first address this question from a philosophical point of view and then I try to find the

Amir Ramezani Dooraki
Department of Artificial Intelligence, Faculty of Computer Science & Information Technology, University of Malaya
E-mail: amir.ramezani.my@gmail.com

machine learning counterpart algorithms or in another word the way that a particular learning method is implemented in machines. Further, I try to discuss about the most important of learning method, considering this paper point of view, and the techniques and methods useful for its implementation in machines.

2 Learning Methodology

"By three methods we may learn wisdom: First, by reflection, which is the noblest; Second, by imitation, which is easiest; and third by experience, which is the bitterest"; Confucius.

It is possible to find similar terminologies to each of the third ways of learning wisdom according to Confucius in machine learning field. "Third by experience, which is the bitterest": this is similar to 'the trial and error part of the reinforcement learning framework'. "Second, by imitation, which is easiest", this is similar to 'learning from demonstration or imitation learning in computer science'. Finally, "First, by reflection, which is the noblest", this important way of learning wisdom ends up creating new ideas and shaping new theories as a result of serious thought or consideration considering a person's ideological framework and according to what that he is learned through the Second and Thirds methods: imitation and experience.

By looking at the background of machine learning algorithms, one can see that supervised learning, unsupervised learning and reinforcement learning are studied and used by researchers for long time and in several cases, and they are still some of the important categories of learning in machines. Nonetheless, it is also possible to categorize them as following:

- Learning by Experience
  - Reinforcement Learning
  - Unsupervised Learning
- Learning by Imitation
  - Learning From Demonstration
    - Pure Classification
    - Inverse Reinforcement Learning
  - Supervised Learning
- Learning by Reflection
  - ?

In the rest of this paper, I first explain briefly about the 'learning by experience' and 'learning by imitation' in machine learning field and later focus on the 'learning by reflection' and try to finds some references for it among machine learning algorithms.

3 Learning by Experience

Learning by trial and error (that is experience) is one of the foremost methods of learning specially for the sensorimotor development stage of intelligent creatures where they get master in moving their motors gradually by trying and fixing their errors. "By observing the development stages of an infant in terms of learning to articulate words, one can see that he first discovers how to control phonation, then focuses on vocal variations of unarticulated sounds, and finally automatically discovers and focuses on babbling with articulated protosyllables" [2]. In the following, it is explained how this method is implemented in machine learning community under different titles.

3.1 Reinforcement Learning

"RL algorithms address the problem of how a behaving agent can learn to approximate an optimal behavioral strategy, usually called a policy, while interacting directly with its environment"[3].

Reinforcement learning family of algorithm are important in machine learning community for their high capability in decision making by learning from their own experiences. For an agent to be able to learn in an unknown environment, it should be able to autonomously interact with the environment. The framework of reinforcement learning is able to give the agent such an ability by letting it to take actions and measure the value of each action based on the reward it receives. More actions the agent takes, it gets more information about what is the best move in each state of the environment.
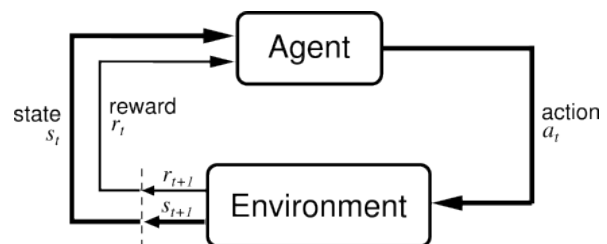

Fig. 1 Reinforcement Learning Framework

Reinforcement learning [4] algorithms can be divided into several groups such as model-based and model-free, temporal difference methods, policy gradient methods and actor critic methods. Furthermore, they can be based on one or multi-step prediction [5]. An interesting trend in these algorithms is the idea of intrinsic moti-

vation that makes the embodiment of qualities such as curiosity in machines possible.

The idea of 'learning by experience' in reinforcement learning frame work can be considered as a Markov Decision Process case. Where an agent interacts with its environment at discrete time steps and at each time step agent is located in a unique state according to its sensory input $s_t \in S$. In each s_t agent can take an action, $a_t \in A$ based on its current policy $\pi(s_t)$. In the next time step, the state of agent changes to $s_{(t+1)} \in S$ based on its previous action and a reward R that it receives from the environment. (Figure 1).

Temporal Difference Approaches; Model-based and Model-less

Temporal difference algorithms are the reinforcement learning algorithms that try to calculate the value of a state or the values of each possible action in that state with attention to the reward received in each state. Furthermore, this information will be used by the agent to use in its policy in which it helps the agent to choose the best action in each state, and as a result the agent finally would be able to maximize its accumulated reward.

Q-Learning; Model-less Method

In many problems, there is no model of the environment. As a result, a model-free off-policy method such as q-learning [5] can be best to suite the scenario where it can be combined with experience replay method [6] for designing more capable learning algorithms. A temporal difference reinforcement learning method can work for example based on an epsilon-greedy policy for action selection. The goal of this kind of policies is to achieve the maximum discounted reward over time and by using an iterative method of selecting the action with maximum value in each state—and updating this value using itself, the maximum action value in the next state and the reward it receives from the environment [4].

Dynamic Programing, Monte-Carlo and SARSA; A Model-based Method

As it can be noticed by looking at the title of a model-based reinforcement learning algorithm, its difference with model-less RL is in having a model of the environment. No matter, how the environment is, either 1D, 2D or 3D environment, or a mathematical formula of the dynamics of the system, a model based algorithm is able to calculate a future state of the environment with attention to the current state and the desired action. A

very good example of a model-based algorithm is the work in [7] where a machine is able to play Go game in a human level and win over many human players.

Function Approximator

Many reinforcement learning algorithms use the mentioned formula in order to find the optimal action value. However, there are two main problems, the lack of generalization and the problem of number of the states. In order to tackle these problems a function approximator an be used in order to find the optimal values of each action or state. As a result of using function approximators, Combination of supervised and unsupervised learning with reinforcement learning can be seen in several works. For example, combination of unsupervised learning (self-organizing map) and reinforcement learning can be seen in several works such as [8] where the input signals of the robot sensors are clustered using a dynamic variation of Kohonen self-organizing map algorithm [9], and each cluster is defined to be an state for the reinforcement learning agent. As a result, the RL agent is able to control the robot movement based on pre-defined goals and rewards.

Further, combination of reinforcement learning algorithm and supervised learning algorithms such as neural network can ends up creating a robust class of algorithm in terms of learning to achieve an optimal behavior similar to a human [10] in playing Atari game by defining a Deep Q-Network which is as a matter of fact combination q-learning and deep learning [11].

Actor Critic Approaches

Based on this view, reinforcement learning methods can be grouped under two categories. Actor-only methods and critic-only methods.

- "Actor-only methods work with a parameterized family of policies. The gradient of the performance, with respect to the actor policies, is directly estimated by simulation, and the parameters are updated in a direction of improvement." [12]
- "Critic-only methods rely exclusively on value function approximation and aim at learning an approximate solution to the bellman equation, which will the hopefully prescribe a near-optimal policy." [12]

Actor-critic methods [13], try to mix the important properties of actor-only and critic-only methods. The actor's policy parameters will be updated with the goal of improving the performance using the value function learned by critic through approximation architecture. "The word actor and critic are synonyms for the policy and action-value function, respectively" [14]. One

of the fresh important works in this area is the deep deterministic policy gradient [15] where it is an actor-critic network that is consist of two multilayer or deep artificial neural network based actor and critic. This algorithm could successfully solve the problem of continuous action control in some games and classic control problems.

Policy Gradient Methods

As mentioned in the previous section, actor-only methods are those in which the policy is parameterized and can be optimized directly. "The benefit of a parameterized policy is that a spectrum of continuous actions can be generated, but the optimization methods used (typically called policy gradient methods) suffer from high variance in the estimate of the gradient, leading to slow learning [12][16][17][18][19]." [14]

3.1.1 Hierarchical Reinforcement Learning Framework

Considering human behavior it "has long been recognized to display hierarchical structure: actions fit together into subtasks, which cohere into extended goal-directed activities" [20].

"Hierarchical reinforcement learning (HRL) decomposes a reinforcement learning problem into a hierarchy of sub problems or subtasks such that higher-level parent-task invoke lower-level child task as if they were primitive actions. A decomposition may have multiple levels of hierarchy" [21]. This "decomposition into subproblems has many advantages. First, policies learned in subproblems can be shared (reused) for multiple parent tasks. Second, the value functions learned in subproblems can be shared, so when the subproblem is reused in a new task, learning of the overall value function for the new task is accelerated. Third, if state abstractions can be applied, then the overall value function can be represented compactly as the sum of separate terms that each depends on only a subset of the state variables. This more compact representation of the value function will require less data to learn, and hence, learning will be faster."[22]

3.2 Unsupervised Learning

Unsupervised learning method cannot be categorized under learning that happens by imitation since there is no output for the input to be imitated by machine, and it is hard to categorize it under learning that happens by experience perhaps because it does not have an attached reward to classify that particular experience as good or bad. However, this paper argues that receiving
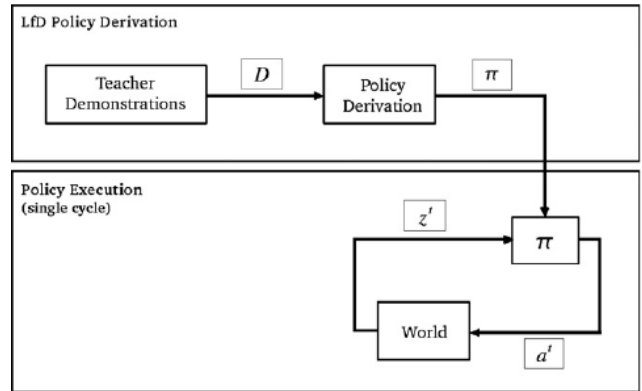

Fig. 2 Learning from Demonstration framework, Adapted from [23]

reward happens through many time steps where each time step can be considered as a new experience for our machine, that is each new sample has a subtle reward (loss) that tries to move a particular input toward a particular class. Thus it can be concluded that it is under the category of 'learning by experience'.

4 Learning by Imitation

The learning capability of human kind develops throughout his life by getting master in 'learning by imitation' (e.g. learning how to write a word from the first grade teacher), a wide spectrum of what we learn after our early ages comes from this type of learning.

4.1 Learning From Demonstration (Apprenticeship Learning)

Learning from Demonstration is a natural and intuitive machine learning approach where an agent that called apprentice tries to learn a behavior from demonstration of another agent called the expert. "We simply show the robot how to achieve a task. This has the immediate advantage of requiring no (or very little) specialized skill or training, and makes use of a human demonstrator's existing procedural knowledge useful to identify which control program to acquire" [24]. A very good example for learning from demonstration, is how to fly an airplane, considering all the different possibilities exist for using the buttons and handles in the cabin of an airplane, the best way to learn how to fly is by observing an expert. Perhaps it is just impossible to try to learn to fly an airplane simply by random pushing of buttons. Looking at the literatures there are at least two methods to deal with this problem of Apprenticeship

Learning (AL); Pure Classification and Inverse Reinforcement Learning (IRL)

4.2 Pure Classification

Pure classification, for instance, the large margin method [25] has several advantages and also some disadvantages that is explained in the following; It is "easy to implement, fast, no need to resolve MDPs, no need to do a choice of features thanks to boosting techniques. However, they do not take into account the structure of the MDP because the temporal structure of the expert trajectories is not used in a pure classification method. To tackle this drawback, the authors of [26] use a kernel-based approach to encapsulate the structure of the MDP into the classifier, which needs the calculation of the MDP metrics and thus the knowledge of the whole dynamics." [27]

4.3 Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL), introduced in [28] and formalized in [29], is related to Apprenticeship Learning (AL) family of algorithm. However, IRL searches to find a reward where this reward should be able to explain the expert behavior and not the expert policy. "The key idea behind IRL is that the reward may be the most succinct hypothesis explaining the expert behavior. Some algorithms [30][31][32] use IRL as an intermediary step to find a policy but other algorithms [33][34] are "pure" IRL algorithms and output a reward. This reward must then be optimized via a direct reinforcement learning algorithm." [27]

4.4 Supervised Learning

This famous method of machine learning helps the machine to learn to imitate that if a particular input received what should be the particular output. As a result, in this paper it is considered to be under the category of 'learning by imitation'.

5 Learning by Reflection

If we describe 'learning by reflection' as the result of serious thought and consideration which can cause for example the creation of a scientific discovery or theory, then it can be explained as a logical outcome of the following cognitive activities:

5.1 Reasoning (Deduction and Induction)

Many of the scientific (e.g computation, biological, social) theories are as a result of a chain of reasonings. Where one can interpret reasoning as an unsupervised or supervised pattern recognition, where for instance by seeing a view a person can recognize it or categorize it as a special event or group, and as a result it can be categorized under the 'learning by experience' or 'learning by imitation'.
Reasonings help us to understand and being able to explain where an incident is coming from or where it will ends up or what is the meaning of a particular perception (e.g. a view, a sound, an smell or a combination of several senses (biological sensors)). Furthermore, following the trajectory of incidents (started by incident 'a') using chain of reasonings one can end up to a particular incident 'b' and form a social or scientific theory, where it begins by incident 'a' and ends up to incident 'b'. For example, a person look is faint, we can reason that its because he is hungry or sick, let say the person is hungry thus he does not have money or is very lazy to buy food, or let say the person is sick so he does not have money to buy food or he got a virus, perhaps his immune system is weak or the virus is new, and this chain of reasoning can continue to reach a particular theory based on the look of a person.

5.2 Imagination

When a person imagines, he creates or shapes a perception in his mind, which has been obtained according to what he has perceived previously (either through experience or imitation) with attention to his current thoughts and ideas; this imagination can be altered or enhanced with-in the framework of his believes (e.g. a scientific belief or a religious belief).
Here, I emphasize that even though machine imagination concept that is discussed in [35] and similar works are interesting but they are not the topic that this paper intends to discuss about. In those works imagination for machine is discussed and explained as a process that convert an input, for example a text, to an image where the process and type of learning can be categorized more under the category of 'learning by imitation' (learning to produce a particular image when input is a particular text or sentence). Nonetheless, what this paper means by "imagination" in a machine is as a matter of fact a process that uses machine memory and knowledge in order to create new perceptions that did not shown to machine before and in order to learn from this new perception.

This paper categorizes imagination into the following categories:

- The first category is a synthesization of world model, based on the past experiences, for example, imagine a person that seen a place before, he is able to imagine this place, or world model, again in his mind and take new actions and calculates new results accordingly. In computer science there are instances of this kind of imagination, for instance SLAM [36], where a machine learn a model of the world and as a result imagines this model and particular outcomes with-in this imaginary model of the world according to particular actions. Model-based reinforcement learning algorithms are among those in which a model of the world can be built, and then refined and completed using experiences and observations. Model-based RL aims to endow an agent "with a model of the world, synthesized from past experience. By using an internal model to reason about the future, here also referred to as imagining, the agent can seek positive outcomes while avoiding the adverse consequences of trial-and-error in the real environment including making irreversible, poor decisions" [37].

  One important consideration in this type of imagination is that it is limited by the model of the world, made by observation, that is, this kind of imagination cannot go beyond the observation of person or machine. In another world, this imagination is limited to experience and observation and it is kind of learning by experience. Thus, perhaps it is possible to think that model-based thinking (method) is a reasoning process which of course can reach to very important discoveries and outcomes.

- In order to explain the second category, first, let's imagine Einstein thinking about time being relative, he is considering that his bus can move faster than a beam of light coming from the clock tower, this idea perhaps is not based on experience or observation (it is more than a model-based method). He was imagining something out of the world-model he experienced by creating at the first point an exception beyond or contradictory to the experiences and observations he had because nobody can move faster than light so it can not be experienced nor it can not be imitated, however, this way of thinking (exceptional or contradictory) paved the way to create a great theory.

  Furthermore, following the chain of reasonings it is possible to reach a point that the person does not have any experience or imitation information about it, for example considering biological advancements and discovery of a method that makes human immortal, the fact of immortal human perhaps is not experienced before, it can be imitated from the movies, but for the first time there was no movie about it, this example is also an imagination that is based on pure imagination contradictory or beyond experiences and imitations.

Is it possible to implement the 'learning by reflection' in machines? As explained in this section, a reflection can be explained as an imagination that is not limited to experience or imitation. Chain of reasonings and limited imagination both can be explained by 'learning by experience' or 'learning by imitation', however, a reflection, needs an elaboration, enhancement, or exception beyond the experienced world-model. This enhancement would happen according to the believe frame-work of the machine or person where this belief frame work can be scientific, religious or spiritual. Further, this enhancement can happen using what learned by imitation or by mixing different experiences into each other and judge the possibility of the result according to the believe framework of the person or machine.

6 Conclusion

In this paper, several ways of learning for a machine —learning by experience, learning by imitation and learning by reflection— inspired by *human ways of learning wisdom explained by Confucius*, introduced and discussed. Further, it explained that two of these methods, 'learning by experience' and 'learning by imitation' are widely researched and implemented in machine learning community under other titles. For example, *reinforcement learning* and *unsupervised learning* are based on 'learning by experience' and *learning from demonstration* and *supervised learning* are based on 'learning by imitation'. Nevertheless, the third way of learning wisdom, 'learning by reflection', is focused less and less implemented as well in machine learning community and algorithms. As a result, this paper discussed about this method of learning and the ways it can be explained through them in minds and machines.

References

1. F. Bellas, P. Caamaño, A. Faiña, R.J. Duro, Evolving Systems **5**(1), 49 (2014). DOI 10.1007/s12530-013-9079-4. URL http://dx.doi.org/10.1007/s12530-013-9079-4
2. C. Moulin-Frier, S.M. Nguyen, P.Y. Oudeyer, Frontiers in Psychology **4**, 1006 (2014). DOI 10.3389/fpsyg.2013.01006. URL http://journal.frontiersin.org/article/10.3389/fpsyg.2013.01006
3. A.G. Barto, *Intrinsic Motivation and Reinforcement Learning* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013), pp. 17–47

4. R.S. Sutton, A.G. Barto, *Introduction to Reinforcement Learning*, 1st edn. (MIT Press, Cambridge, MA, USA, 1998)

5. C.J. Watkins, P. Dayan, Machine Learning **8**(3), 279 (1992). DOI 10.1023/A:1022676722315. URL http://dx.doi.org/10.1023/A:1022676722315

6. L.J. Lin, Reinforcement learning for robots using neural networks. Ph.D. thesis, Pittsburgh, PA, USA (1992). UMI Order No. GAX93-22750

7. D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Nature **529**(7587), 484 (2016). URL http://dx.doi.org/10.1038/nature16961. Article

8. J.d.R. Millán, D. Posenato, E. Dedieu, Machine Learning **49**(2), 247 (2002). DOI 10.1023/A:1017988514716. URL http://dx.doi.org/10.1023/A:1017988514716

9. T. Kohonen, M.R. Schroeder, T.S. Huang (eds.), *Self-Organizing Maps*, 3rd edn. (Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001)

10. V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, et al., Nature **518**(7540), 529 (2015)

11. G.E. Hinton, Trends in Cognitive Sciences **11**(10), 428 (2007). DOI 10.1016/j.tics.2007.09.004. URL http://dx.doi.org/10.1016/j.tics.2007.09.004

12. V.R. Konda, J.N. Tsitsiklis, SIAM J. Control Optim. **42**(4), 1143 (2003). DOI 10.1137/S0363012901385691. URL https://doi.org/10.1137/S0363012901385691

13. A.G. Barto, R.S. Sutton, C.W. Anderson, IEEE Transactions on Systems, Man, and Cybernetics **SMC-13**(5), 834 (1983). DOI 10.1109/TSMC.1983.6313077

14. I. Grondman, L. Busoniu, G.A.D. Lopes, R. Babuska, Trans. Sys. Man Cyber Part C **42**(6), 1291 (2012). DOI 10.1109/TSMCC.2012.2218595. URL http://dx.doi.org/10.1109/TSMCC.2012.2218595

15. T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, CoRR **abs/1509.02971** (2015). URL http://arxiv.org/abs/1509.02971

16. J.A. Boyan, Machine Learning **49**(2-3), 233 (2002). DOI 10.1023/A:1017936530646

17. J. Baxter, P.L. Bartlett, **15**, 319 (2001)

18. S. Richter, D. Aberdeen, Y. Jin, Advances in Neural Information Processing Systems **19**, 1169 (2007)

19. H.R. Berenji, D. Vengerov, IEEE Transactions on Fuzzy Systems **11**(4), 478 (2003). DOI 10.1109/TFUZZ.2003.814834

20. A. Solway, C. Diuk, N. Crdova, D. Yee, A.G. Barto, Y. Niv, M.M. Botvinick, PLoS Computational Biology **10**(8) (2014). DOI 10.1371/journal.pcbi.1003779

21. B. Hengst, *Hierarchical Reinforcement Learning* (Springer US, Boston, MA, 2010), pp. 495–502

22. T.G. Dietterich, CoRR **cs.LG/9905014** (1999). URL http://arxiv.org/abs/cs.LG/9905014

23. B.D. Argall, S. Chernova, M. Veloso, B. Browning, Robotics and Autonomous Systems **57**(5), 469 (2009). DOI http://dx.doi.org/10.1016/j.robot.2008.10.024

24. G. Konidaris, S. Kuindersma, R. Grupen, A. Barto, International Journal of Robotics Research **31**(3), 360 (2012). URL http://ijr.sagepub.com/content/31/3/360

25. N. Ratliff, J.A. Bagnell, S.S. Srinivasa, Proceedings of the 2007 7th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS 2007 pp. 392–397 (2008). DOI 10.1109/ICHR.2007.4813899

26. F.S. Melo, M. Lopes, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6322 LNAI**(PART 2), 385 (2010)

27. B. Piot, M. Geist, O. Pietquin, in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2014), AAMAS '14, pp. 1249–1256. URL http://dl.acm.org/citation.cfm?id=2615731.2617447

28. S. Russell, Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT) pp. 101–103 (1998). DOI 10.1145/279943.279964

29. A. Ng, S. Russell, Proceedings of the Seventeenth International Conference on Machine Learning **0**, 663 (2000). DOI 10.2460/ajvr.67.2.323

30. P. Abbeel, A.Y. Ng, Proceedings of the 21st International Conference on Machine Learning (ICML) pp. 1–8 (2004). DOI 10.1145/1015330.1015430

31. G. Neu, C. Szepesvári, Machine Learning **77**(2-3), 303 (2009). DOI 10.1007/s10994-009-5110-1

32. U. Syed, R.E. Schapire, Advances in Neural Information Processing Systems 20 **20**, 1 (2008)

33. A. Boularias, J. Kober, J. Peters, J.K. A. Boularias, Icaps **15**, 20 (2011)

34. E. Klein, M. Geist, B. Piot, O. Pietquin, 2Advances in Neural Information Processing Systems pp. 1–9 (2012)

35. H. Kato, T. Harada, CoRR **abs/1505.05190** (2015). URL http://arxiv.org/abs/1505.05190

36. H. Durrant-Whyte, T. Bailey, IEEE Robotics Automation Magazine **13**(2), 99 (2006). DOI 10.1109/MRA.2006.1638022

37. T. Weber, S. Racaniere, D.P. Reichert, L. Buesing, A. Guez, D.J. Rezende, A.P. Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu, P. Battaglia, D. Silver, D. Wierstra, CoRR **abs/1707.06203** (2017). URL http://arxiv.org/abs/1707.06203