A critic-critic architecture to combine reinforcement and supervised learnings

Fabien Montagne MONTAGNE@LIL.UNIV-LITTORAL.FR
Samuel Delepoulle DELEPOULLE@LIL.UNIV-LITTORAL.FR
Laboratoire d'Informatique du Littoral (LIL), Université du Littoral Côte d'Opale, UPRES-JE 2335, B.P. 719, 62228 Calais Cedex, France

Philippe Preux PPREUX@UNIV-LILLE3.FR
Groupe de Recherche en Apprentissage Automatique (GRAPPA), Université de Lille 3, UPRES-EA 3588, B.P. 149, 59653 Villeneuve d'Ascq Cedex, France

In real life, learning is greatly speeded-up by the intervention of a teacher who gives examples, or shows, how to perform a certain task. In all this abstract, we let apart structural simplifications of the problem by the designer which to not deal explicitely with learning. The intervention of the teacher can be realized in different ways: verbal explanation, demonstration, guidance, shaping the behavior, ... see e.g. (Jordan, 1986; Gaussier et al., 1997; Hugues & Drogoul, 2001; Rosenstein & Barto, 2002).

Conceptually, this means combining reinforcement learning with supervised learning. In this work, we focus on the guidance technique in which we virtually take the learner's hand and make it perform the task, or a part of it. The examples are trajectories in the state space. Having a set of examples, many ways of performing such a combination may be considered. Here, we wish to provide some guidance to a reinforcement learner and take advantage of a number of its properties: use the guidance as a help rather than a strict order; improve the examples that have been given as a guidance; adapt whenever the environment changes; generalize as much as possible from the set of examples. The latter feature is related to the use of a relevant architecture to store current estimates of the value of states. The other three features are related to the trade-off between exploration and exploitation.

As reinforcement learners, we consider those known as temporal difference methods (TD) (Sutton & Barto, 1998). At a certain time $t \in \mathbb{N}$, the learner perceives the state of its environment $s_t \in \mathcal{S}$, chooses an action to perform $a_t \in \mathcal{A}$, and emits it. Then, the environment provides an immediate return $r_t \in \mathbb{R}$, as well as the new state s_{t+1}. The TD algorithm aims at learning a policy, that is, a mapping from $\mathcal{S} \times \mathcal{A} \mapsto [0,1]$ that indicates the probability to emit a certain action in a certaion state so as to maximize the expected return $R_t = \sum_{k=0}^{k=T} \gamma^{t+k} r_{t+k}$, where T is the time at which a certain goal state is reached (possibly $T = +\infty$). To do this, a TD learner estimates the value of each state $V(s) = E(R_t \mid s_t = s)$, where $E(.)$ denotes the mathematical expectation. The value allows the TD learner to estimate the expected immediate return and it is then able to compare it to the actual immediate return. Then, the TD learner updates its estimate of the value using the difference between the expected return and the actual return (TD error).

After having described the general context, let us be more specific about our point. As we said earlier, when the learner chooses the action to perform in its current state, it has to balance the exploration and the exploitation. As pointed out earlier, this balance has an important and complex role (Thrun, 1992). The selection of action strategy in the context of a TD learner receiving some guidance from an external teacher is thus the topic of this abstract for which we propose a novel critic-critic architecture.

One common issue in learning is overfitting. The TD-learner uses a neural network as the value function approximator and is based on (Tesauro, 1992) adapted to a continuous space as in (Coulom, 2002). The TD learner should generalize from the example trajectories. In a realistic application, the definition of "states" (i.e. the definition of $\mathcal{S}$) is generally not obvious: if the state is too rich, the learner may not be able to generalize to other states; too poor, it might not be able to solve the task. Then, we propose to use two representation levels: one using coarse grain, the other one using finer grain. Considering the "real" state is a real-valued vector S, the fine grain representation is this vector, while the coarse grain representation is a projection of this vector in a subspace s.

The coarse grain level learns a value function $V(s)$ while the finer grain level learns a confidence $C(S)$ associated to $V(s)$. The idea is to use this confidence to balance exploration and exploitation: if confidence is high, then exploit by choosing preferably the greedy action; if confidence is low, then explore by choosing an other action. The coarse grain level produces generalization while the finer grain level corrects the former. Bringing on the idea of the actor-critic architecture, we propose to realize this idea with an architecture having no actor, but two critics, one critic for each level. The coarse grain critic (CC) provides an estimate of the value of states (V(s)). Then, the fine grain critic (CF) assigns a certain level of confidence the estimate can be attributed (C(S)). Note that by default, the fine grain critic trusts the coarse grain critic. The whole algorithm is sketched below.

Algorithm 1 *The critic-critic algorithm*:

```
1- Initialization of the learner
Use the example trajectories to estimate V($s$) in CC
2- Unsupervised learning
$t = 0$, $S_t$= initial state, $\forall S$ C($S$)$\simeq$0.5,
eligibility trace ET = $\emptyset$
3- Find a trajectory
while $S_t$ non terminal do
  $\mathscr{S}$=$\{S$, $S$ reachable from $S_t$ at $t+1\}$
  for $S \in \mathscr{S}$ do
    $s$=projection of $S$ in the subspace of CC
    T($S$)=C($S$)$\times$V($s$)
  end for
  $S_t$=$\underset{S\in\mathscr{S}}{\text{argmax}}$(T($S$))
  if $S_t \notin$ ET then Add $S_t$ to ET
  Devaluate C($S_t$) : C($S_t$) = C($S_t$) $\times \alpha$ , $\alpha \in ]0,1[$
end while
for all $S \in$ ET do
  Increase C($S$) : C($S$) = $\frac{C(S)}{\alpha} + \frac{1}{|ET|}$
end for
```

For example, we may consider the following problem. Assume that a mobile agent has to reach a certain location using as few energy as possible, given a certain initial amount. The agent is characterized by its current position and velocity in 3 dimensions and some other properties. The value of a state characterizes the cost to reach the target. The fine grain state S is made of all these data whereas the coarse grain state s may be restricted to the x and y positions and the velocity along z. This definition is based on a prior analysis of the task which revealed that these 3 data are the most important in the value of a state. Stated otherly, these 3 data being fixed, the variability of the value when the other data contributing to a fine grain state is small. So the value assigned to a state s made of these 3 data has a certain level of confidence according to the various S which are projected onto s.

We are currently experimenting and assessing the critic-critic architecture. A forthcoming improvement of the critic-critic algorithm consists in merging the two phases of supervised and non supervised learning, so as to perform incremental learning. Being common issues in supervised learning, we also aim at characterizing the number of examples that are required to obtain good generalization. We definitely think that the combination of reinforcement learning with supervised learning can be fruitfully adapted to a large spectrum of problems.

References

Clouse, J. (1997). The role of training in reinforcement leanring. In J. Donahoe and V. P. Dorsel (Eds.), *Neural-networks lmodels of cognition*, 422–435. Elsevier.

Coulom, R. (2002). *Apprentissage par renforcement utilisant des reseaux de neuronnes, avec des applications au controle moteur*. Doctoral dissertation, Institut national polytechnique de Grenoble.

Dorigo, M., & Colombetti, M. (1994). Robot shaping: developing autonomous agents through learning. *Artificial Intelligence*, *71*, 321–370.

Gaussier, P., Moga, S., Banquet, J., & Quoy, M. (1997). From perception-action loop to imitation processes: a bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, *1*.

Gullapalli, V. (1997). Reinforcement leanring of complex behavior through shaping. In J. Donahoe and V. P. Dorsel (Eds.), *Neural-networks lmodels of cognition*, 302–314. Elsevier.

Hugues, L., & Drogoul, A. (2001). Shaping of robot behaviors by demonstrations. *Proc. of the first Int'l Workshop on Epigenetic Robotics: Modeling cognitive development in robotic systems.*

Jordan, M. (1986). Attractor dynamics and parallelim in a connectionnist sequential machine. *Proc. of the 1986 Cognitive Science Conference* (pp. 531–546).

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning, and teaching. *Machine Learning*, *8*, 293–322.

Rosenstein, M., & Barto, A. (2002). *Supervised learning combined with an actor-critic architecture* (Technical Report). Dpt. of Computer Science, Univ. of Massachussets, Amherst, MA, USA.

Sutton, & Barto (1998). *Reinforcement learning*. MIT Press.

Tesauro, G. (1992). Practical issues in temporal difference learning. *Machine Learning*, *8*, 257–277.

Thrun, S. (1992). The role of exploration in learning control. In *Handbook of intelligent control: neural, fuzzy, and adaptive approach*, 527–559. Van Nosdtrand Reinhold.