what it has already learned. But what should such an agent do when it inevitably runs out of resources? One possible solution is to prune away less useful skills and knowledge, which is difficult if these are closely connected to each other in a network of complex dependencies. The approach I advocate in this talk is to give the agent at the outset all the computational resources it will ever have, such that continual learning becomes the process of continually reallocating those fixed resources. I will describe how an agent's policy can be broken into many pieces and spread out among many computational units that compete to represent different parts of the agent's policy space. These units can then be arranged across a lower-dimensional manifold according to those similarities, which results in many advantages for the agent. Among these advantages are improved robustness, dimensionality reduction, and an organization that encourages intelligent reallocation of resources when learning new skills.

3.24 Multi-objective Reinforcement Learning

Manuela Ruiz-Montiel (University of Malaga, ES)

Joint work of Ruiz-Montiel, Manuela; Mandow, Lawrence; Pérez-de-la-Cruz, José-Luis
Main reference M. Riuz-Montiel, L. Mandow, J. L. Pérez-de-la-Cruz, "PQ-Learning: Aprendizaje por Refuerzo Multiobjetivo," in Proc. of the XV Conference of the Spanish Association for Artificial Intelligence (CAEPIA'13), pp. 139–148, 2013.
URL http://www.congresocedi.es/images/site/actas/ActasCAEPIA.pdf

In this talk we present PQ-learning, a new Reinforcement Learning (RL)algorithm that determines the rational behaviours of an agent in multi-objective domains. Most RL techniques focus on environments with scalar rewards. However, many real scenarios are best formulated in multi-objective terms: rewards are vectors and each component stands for an objective to maximize. In scalar RL, the environment is formalized as a Markov Decision Problem, defined by a set S of states, a set A of actions, a function $P_{sa}(s')$ (the transition probabilities) and a function $R_{sa}(s')$ (the obtained scalar rewards). The problem is to determine a*policy* $\pi : S \to A$ that maximizes the *discounted accumulated reward* $R_t = \Sigma_{k=0}^{\infty} \gamma^k r_{t+k+1}$.E.g., Q-learning [1] is an algorithm that learns such policy. It learns the scalar values $Q(s, a) : S \times A \to \mathbb{R}$, that represent the expected accumulated reward when following a given policy after taking a in s. The selected action a in each state is given by the expression $argmax_a Q(s, a)$. In the multi-objective case the rewards are vectors $\overrightarrow{r} \in \mathbb{R}^n$, so different accumulated rewards cannot be totally ordered; $\overrightarrow{v}$ dominates $\overrightarrow{w}$ when $\exists i : v_i > w_i \wedge \nexists j : v_j < w_j$. Given a set of vectors, those that are not dominated by any other vector are said to lie in the *Pareto front.* We seek the set of policies that yield non-dominated accumulated reward vectors. The literature on multi-objective RL (MORL) is relatively scarce (see Vamplew et al. [2]). Most methods use preferences (lexicographic ordering or scalarization) allowing a total ordering of the value vectors, and approximate the front by running a scalar RL method several times with different preferences. When dealing with non-convex fronts, only a subset of the solutions is approximated. Some multi-objective dynamic programming (MODP) methods calculate all the policies at once, assuming a perfect knowledge of $P_{sa}(s')$ and $R_{sa}(s')$. We deal with the problem of efficiently approximating all the optimal policies at once, without sacrificing solutions nor assuming a perfect knowledge of the model. As far as we know, our algorithm is the first to bring these featurestogether. As we aim to learn a set of policies at once, Q-learning is apromising starting point, since the policy used to interact with theenvironment is not the same that is learned. At each step, Q-learning shifts the previous estimated Q-value towards its new estimation: $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma max_{a'} Q(s',a'))$. In PQ-learning, Q-values are sets of vectors, so the max operator is replaced by $ND(\bigcup_{a'} Q(s',a')))$, where ND calculates the Pareto front. A naive approach to perform the involved set addition is a pairwise summation (imported from MODP methods), but it leads to an uncontrolled growth of the sets and the algorithm becomes impractical, as it sums vectors that correspond to different action sequences. The results of these mixed sums are useless when learning deterministic policies, because two sequences cannot be followed at once. We propose a controlled set addition that only sums those pairs of vectors that correspond to useful action sequences. This is done by associating each vector $\overrightarrow{q}$ with two data structures with information about the vectors that (1) have been updated by $\overrightarrow{q}$ and (2) have contributed to its value. In this talk we describe in detail the application ofPQ-learning to a simple example, and the results that the algorithm yields when applied to two problems of a benchmark [2]. It approximates all the policies in the true Pareto front, as opposed to the naive approach, that produces huge fronts with useless values that dramatically slow down the process.[1]

References

1 C.J. Watkins, *Learning From DelayedRewards.* PhD Thesis, University of Cambridge, 1989.

2 P. Vamplew et al., *Empirical EvaluationMethods For Multiobjective Reinforcement Learning*, in Machine Learning84(1-2) pp. 51-80, 2011.

3.25 Recent Advances in Symbolic Dynamic Programming for Hybrid MDPs and POMDPs

Scott Sanner (NICTA – Canberra, AU)

Joint work of Sanner, Scott; Zamani, Zahra

Many real-world decision-theoretic planning problems are naturallymodeled using mixed discrete and continuous state, action, and observation spaces, yet little work has provided *exact* methodsfor performing exact dynamic programming backups in such problems. This overview talk will survey a number of recent developments in the exact and approximate solution of mixed discrete and continuous (hybrid) MDPs and POMDPs via the technique of symbolic dynamic programming (SDP) as covered in recent work by the authors [1, 2, 3, 4].

References

1 S. Sanner, K. V. Delgado, and L. Nunes de Barros. Symbolic dynamic programming for discrete and continuous state MDPs. In *In Proc. of the 27th Conf. on Uncertainty in Artificial Intelligence (UAI-11)*, Barcelona, Spain, 2011.

2 Z. Zamani, S. Sanner, K. V. Delgado, and L. Nunes de Barros. Robust optimization for hybrid mdps with state-dependent noise. In *Proc. of the 23rd International Joint Conf. on Artificial Intelligence (IJCAI-13)*, Beijing, China, 2013.

[1] This work is partially funded by: grant TIN2009-14179 (SpanishGovernment, Plan Nacional de I+D+i) and Universidad de Málaga, Campusde Excelencia Internacional Andalucía Tech. Manuela Ruiz-Montiel is funded by the Spanish Ministry of Education through the National F.P.U. Program.