Late-Breaking Developments in the Field of Artificial Intelligence
Papers Presented at the Twenty-Seventh AAAI Conference on Artificial Intelligence

An Ensemble of Linearly Combined Reinforcement-Learning Agents

Vukosi Marivate* and Michael Littman**
Department of Computer Science, Rutgers University
*Department of Computer Science, Brown University
vukosi@cs.rutgers.edu, mlittman@cs.brown.edu

Abstract

Reinforcement-learning (RL) algorithms are often tweaked and tuned to specific environments when applied, calling into question whether learning can truly be considered autonomous in these cases. In this work, we show how more robust learning across environments is possible by adopting an ensemble approach to reinforcement learning. Our approach learns a weighted linear combination of Q-values from multiple independent learning algorithms. In our evaluations in generalized RL environments, we find that the algorithm compares favorably to the best tuned algorithm. Our work provides a promising basis for further study into the use of ensemble methods in RL.

The task of creating a single reinforcement-learning (RL) agent that can learn in many possible environments without modification is not a simple one. It is typical for algorithm designers to modify state representations, learning protocols, or parameter values to obtain good performance on novel environments. However, the more problem-specific tuning needed, the less "autonomous" an RL system is, eroding some of the value of RL systems in practice. Often, the process of tuning itself requires agents to repeatedly learn and relearn in the target environment—an approach that simply cannot be used in practice.

Across a wide range of computational domains, ensemble learning methods have proven extremely valuable for reliably tackling complex problems. Ensemble (or sometimes modular or portfolio) methods harness multiple, perhaps quite disparate, algorithms for a problem class to greatly expand the range of specific instances that can be addressed. They have emerged as state-of-the-art approaches for word sense disambiguation (Florian and Yarowsky 2002), crossword solving (Littman, Keim, and Shazeer 2002), satisfiability testing (Xu, Hoos, and Leyton-Brown 2010), movie recommendation (Bell, Koren, and Volinsky 2010) and question answering (Ferrucci et al. 2010). We believe the success of ensemble methods on these problems stems from the fact that they can deal with a range of instances that require different low-level approaches. RL instances share this attribute, suggesting that an ensemble approach could be valuable there as well.

TD Combination of RL Agents

In this work, we present an approach to ensemble-based RL using a linear Temporal Difference (TD) learning algorithm as a meta learner to combine the value estimates from multiple base RL algorithm agents. Our approach goes beyond earlier efforts in ensemble RL (Wiering and van Hasselt 2008) in that we develop a fusion method that is adjusted given the performance of the base agents in the ensemble instead of combining low-level agents according to a fixed rule. In the ensemble classifier approach, given m classifiers, each classifier i has a prediction $d_{i,j}(\mathbf{x})$ as to whether the data point $\mathbf{x}$ belongs to class ω_j. The final prediction of the meta learner, $\mu_j(\mathbf{x})$, for class ω_j is $\mu_j(\mathbf{x}) = \sum_{i=1}^{m} w_{i,j} d_{i,j}(\mathbf{x})$. The supervised meta learner uses held out labeled training data to learn the combination weights w_i for each base classifier. Whereas supervised classifiers map instances to labels, in a value-function-based setting, RL algorithms map states and actions to action values (the so-called Q-function).

Using the supervised ensemble weighted learning as a guide, we can develop a parallel approach in which separate RL algorithms (base agents) create their own Q functions. The RL meta learner then estimates the environment's Q-function via a weighted linear combination of the Q-values learned by the base agents. The final Q-value given n RL base agents is $Q_{\mathbf{W}}(s_t, a_t) = \sum_{k=1}^{n} w_k Q_k(s_t, a_t)$, where w_k are the weights and $Q_k(s, a)$ is the estimated Q-value of state s and action a for RL base agent k. The RL meta learner learns the weights w_k for each base agent. Given that labeled examples are not available in the RL setting, another error metric needs to be used. In TD-based algorithms (Sutton 1988), the natural error metric is the Bellman error,

$$E_{RL,\mathbf{w}} = \sum_{t} [r(s_{t+1}, a_{t+1}) + \gamma Q_{\mathbf{W}}(s_{t+1}, a_{t+1}) - Q_{\mathbf{W}}(s_t, a_t)]^2, \quad (1)$$

where $r(s_{t+1}, a_{t+1})$ is the reward observed from being in state s_{t+1} and performing action a_{t+1} and γ is the discount factor. An advantage of this error metric is that it does not require labeled examples. This formulation, arrived at by translating standard linear ensemble methods to the RL setting, is an exact match for the problem solved by linear TD methods. The twist is that the role of "state" in this formulation is the Q-value estimates produced by the base agents.

With that substitution in place, any existing TD method can be applied to learn weights for the meta learner.

Given that both the base agents and the meta learner need to adapt, we run learning in two stages. First, the base agents are trained on the environment in question by themselves, then they are frozen and then the meta learner adapts its weights to combine the Q-values of the base agents. We have experimented with adapting the meta learner and base agents simultaneously, but the results were less stable.

As the meta-learner searches for the best linear combination of the base learner Q-values, using them as features, convergence guarantees are similar to those of other linear TD learning algorithms (Tsitsiklis and Van Roy 1997). With the above description of the combination of base agents, we can view each base agent's Q-value as a feature for the meta learner that is dependent on a state and action pair, (s, a). The two-stage meta learner is a least squares algorithm (Boyan 2002) that minimizes Equation 1.

Ensemble Approach to Solving MDPs

To assess the ensemble approach in an RL setting, we carried out our evaluation in the generalized MDPs framework (Whiteson et al. 2009). Instead of creating an RL algorithm by iteratively tuning learning parameters on a fixed environment, the generalized MDP perspective is to imagine that MDPs are drawn from a *generalized environment* $G : \Theta \rightarrow [0, 1]$ from which individual MDPs can be sampled. A learning system can draw MDPs from this distribution for "training". At the end of the training period, a concrete RL algorithm is produced and it is evaluated by running it in fresh MDPs drawn from G. This approach to evaluation is designed to encourage the development of algorithms that do not "overfit" and can thus generalize better across environments.

The environment in our experiments is the classic mountain-car environment. The mountain-car environment was generalized by adjusting observations from the environment (noisy observations), as well the outcomes of the actions taken (stochastic actions). The implementation of the generalized mountain-car environment was taken from the 2008 RL competition (Whiteson, Tanner, and White 2010).

A standard approach would be to tune the learning parameters of a specific algorithm to perform well, on average, on the training MDPs. The output of the learning system, using the training MDPs, would be an RL learner with tuned parameters. In the ensemble approach, one would create a diverse set of learners that would be tuned for generalization on the training MDPs. These learners would then be used to tackle the test MDPs. Thus, the output of the learning system would be an ensemble RL learner comprising of a diverse set of parameters.

Using this approach, we had access to 10 training MDPs. For each of the training MDP, 10 learners with different parameters were trained and evaluated. The parameters of the top learner from each of the MDPs were compared against each other for diversity. Some of the parameters were equivalent and thus only 4 parameters were used for the base learners of the meta learner. The candidate parameter comparison for diversity can be accomplished by a clustering

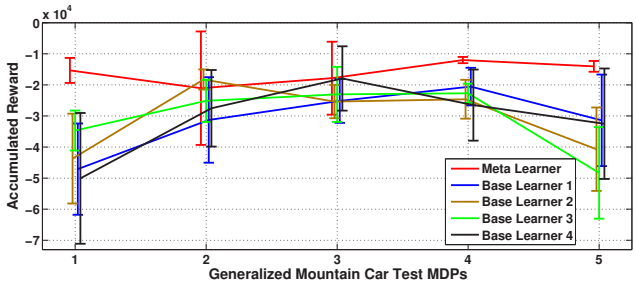

Figure 1: A meta-learner that combines the results of multiple RL algorithms achieves higher reward across a collection of environments than individual RL algorithms.

algorithm but in this case was done manually. The RL base learners created using the 4 parameters were also evaluated individually on the test MDPs.

For evaluation, the learners (4 individual Q-Learning base learners) and the metalearner (Least Squares Temporal Difference Learning) had 1000 training episodes on each test MDP. The meta learner equally allocated episodes of interaction between itself and each of its base learners. Evaluation was on the last 50 learning episodes. The meta learner's exploration rate (ε) was fixed at 0.05 for all tests. The accumulated rewards for each test MDP configuration were averaged over 5 runs and are shown in Figure 1. The meta learner consistently performs better than the best single base learner. Further, if a single set of parameters was used across all the MDPs, the resulting algorithm would not have performed as well as the meta-learner.

Further Work

We have presented an ensemble RL algorithm that weighs and combines estimates from multiple RL algorithms. Using a TD algorithm, the high-level ("meta") learner is able to identify how to weigh and combine low-level (base) agents so as to obtain high returns. Since the meta learner weighs and combines Q values, other algorithms for the base agents can be added/substituted as long as they estimate Q values. Model-based RL algorithms such as RMax (Brafman and Tennenholtz 2003) could be substituted for the base agents as they compute Q-functions indirectly. The meta learner's TD algorithm can also be substituted with other more efficient algorithms. For example, we could have the meta learner be a selective learner by using LARS-TD (Kolter and Ng 2009).

There is an opportunity to improve some of the state-of-the-art algorithms in RL by using them in an ensemble setting, since doing so has been shown to reduce error. The effects of varying parameters, such as the discount factor, are still not clear could provide better insight in diversity creation for the ensemble RL setting. Furthermore, investigating how the size of ensemble can be dynamically changed—removing or adding base agents—is another avenue worth investigating. These promising initial results indicate that reinforcement-learning algorithms can benefit substantially from the ensemble approach.

References

Bell, R. M.; Koren, Y.; and Volinsky, C. 2010. All together now: A perspective on the Netflix Prize. *Chance* 24–29.

Boyan, J. 2002. Technical update: Least-squares temporal difference learning. *Machine Learning* 233–246.

Brafman, R., and Tennenholtz, M. 2003. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research* 213–231.

Ferrucci, D.; Brown, E.; Chu-Carroll, J.; Fan, J.; Gondek, D.; Kalyanpur, A.; Lally, A.; Murdock, J.; Nyberg, E.; Prager, J.; et al. 2010. Building watson: An overview of the deepqa project. *AI Magazine* 59–79.

Florian, R., and Yarowsky, D. 2002. Modeling consensus: classifier combination for word sense disambiguation. In *Proceedings of the ACL conference on Empirical methods in natural language processing*, EMNLP '02, 25–32.

Kolter, J., and Ng, A. 2009. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 521–528.

Littman, M.; Keim, G.; and Shazeer, N. 2002. A probabilistic approach to solving crossword puzzles. *Artificial Intelligence* 23–55.

Sutton, R. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 9–44.

Tsitsiklis, J., and Van Roy, B. 1997. An analysis of temporal-difference learning with function approximation. *Automatic Control, IEEE Transactions on* 674–690.

Whiteson, S.; Tanner, B.; Taylor, M.; and Stone, P. 2009. Generalized domains for empirical evaluations in reinforcement learning. In *Proceedings of the 4th workshop on Evaluation Methods for Machine Learning at ICML-09, Montreal, Canada*.

Whiteson, S.; Tanner, B.; and White, A. 2010. The reinforcement learning competitions. *AI Magazine* 31:81–94.

Wiering, M., and van Hasselt, H. 2008. Ensemble algorithms in reinforcement learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 930–936.

Xu, L.; Hoos, H.; and Leyton-Brown, K. 2010. Hydra: Automatically configuring algorithms for portfolio-based selection. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 210–216.