Ensembles of Shapings

Tim Brys
Vrije Universiteit Brussel
Brussels, Belgium
timbrys@vub.ac.be

Anna Harutyunyan
Vrije Universiteit Brussel
Brussels, Belgium
aharutyu@vub.ac.be

Matthew E. Taylor
Washington State University
Pullman, WA
taylorm@eecs.wsu.edu

Ann Nowé
Vrije Universiteit Brussel
Brussels, Belgium
anowe@vub.ac.be

Abstract

Many reinforcement learning algorithms try to solve a problem from scratch, i.e., without *a priori* knowledge. This works for small and simple problems, but quickly becomes impractical as problems of growing complexity are tackled. The reward function with which the agent evaluates its behaviour often is sparse and uninformative, which leads to the agent requiring large amounts of exploration before feedback is discovered and good behaviour can be generated. Reward shaping is one approach to address this problem, by enriching the reward signal with extra intermediate rewards, often of a heuristic nature. These intermediate rewards may be derived from expert knowledge, knowledge transferred from a previous task, demonstrations provided to the agent, etc. In many domains, multiple such pieces of knowledge are available, and could all potentially benefit the agent during its learning process. We investigate the use of ensemble techniques to automatically combine these various sources of information, helping the agent learn faster than with any of the individual pieces of information alone. We empirically show that the use of such ensembles alleviates two tuning problems: (1) the problem of selecting which (combination of) heuristic knowledge to use, and (2) the problem of tuning the scaling of this information as it is injected in the original reward function. We show that ensembles are both robust against bad information and bad scalings.

Keywords: Reward Shaping; Ensembles

Acknowledgements

Tim Brys is funded by a Ph.D grant of the Research Foundation Flanders (FWO). Anna Harutyunyan is supported by the IWT-SBO project MIRAD (grant nr. 120057). This work was supported in part by NSF IIS-1149917.

1 Motivation

With many reinforcement learning algorithms taking a *tabula rasa* approach, their sample complexity is often prohibitively high to be useful in realistic settings. In other words, they require too many experiences, too much 'trial-and-error,' before reaching a desirable level of performance. Imagine a task where the agent only receives positive reward after a very specific, complex sequence of actions has been executed (e.g., the 'combination lock' problem [1]). If the goal of the task is to execute this sequence of actions, then the reward function perfectly encodes this task. But, due to its sparsity, it will also likely result in very slow learning. A lot of research has therefore focused on speeding up these reinforcement learning algorithms by steering their exploration based on expert knowledge [2], knowledge transferred from previous tasks [3, 4], provided demonstrations [5, 6], human advice [7, 8, 9], abstract knowledge learned during learning [10], etc. In many cases, one has several such pieces of information available, e.g., several heuristic rules can be devised, or multiple source tasks are available to transfer from, etc. Problems a system designer is then faced with are (1) how to include this knowledge in the learning process, and (2) how to combine the various pieces of knowledge in an optimal way. More often than not, the system designer would start his own trial-and-error process of trying out combinations and tuning their parameters. This tuning process often requires many more experiences than are gained in the end by using the best performing combination. In reality, we would like an off-the-shelf solution that we can supply with different forms of information, and that can combine these automatically in a near-optimal way.

Our contribution towards this goal consists of injecting the information in the learning process through an approach called reward shaping, and using ensemble techniques to automatically and robustly combine the various pieces of information supplied.

2 Reward Shaping

Recall the example above, where the environment's reward is only positive when the required sequence of actions has been executed. If we can find a way to provide positive feedback for each step in this sequence of actions, the task will become easily learnable for the agent, as its behaviour is reinforced at every step. Of course, typically we do not know the solution beforehand, and can only provide information of a heuristic nature, i.e., rules of thumb that provide general guidelines, but are not perfect in every situation.

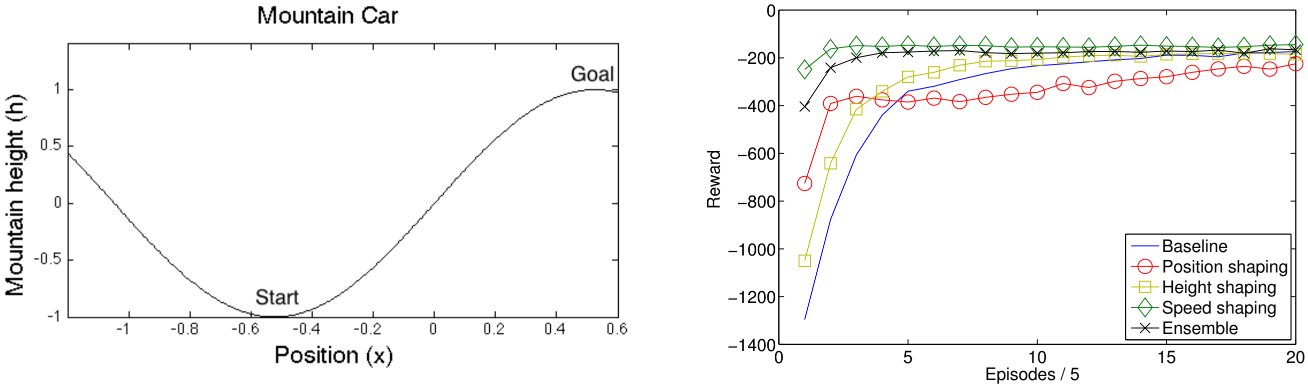
The idea behind reward shaping is to harness such information to enrich the environment's sparser reward and thus provide faster, more informative feedback for the agent's behaviour. The agent is supplied with an extra reward signal F that is added to the environment's reward R, making the agent learn on the composite signal $R_F = R + F$. Since the agent's goal is defined by the reward function (solving the task optimally means finding a description of behaviour, i.e., a policy, that achieves the maximum accumulated reward in expectation), changing the reward signal may actually change the task. Ng et al. [2] proved that the only sound way to modify the reward, while guaranteeing that the task's optimal policy does not change, is through potential-based shaping. That is, define a potential function Φ over the state space, and define F as the difference between the potential of states s' and s, given observed transition (s, a, s'):

$$F(s, a, s') = \gamma\Phi(s') - \Phi(s)$$

This formulation preserves the total order over policies, and therefore also the optimality of policies. It has been successfully used to facilitate solving of such complex tasks as RoboCup TakeAway [11], StarCraft [12], Mario [13], helicopter flight [14, 15], etc.

The intuitition behind defining Φ is that states with high potential will be desirable to the agent, i.e., it will be encouraged to explore such states. A good potential function should therefore yield higher and higher potentials as the agent gets closer and closer to states that are desirable with respect to the base reward, thus quickly leading the agent to optimal behaviour. Again, in absence of knowledge of the full solution, we can only use heuristic information when defining Φ. Consider Mountain Car [16], a problem where an underpowered car, starting in the valley between two hills, needs to learn how to drive to the top of one of the hills by driving up and down the opposing hills, thereby building up momentum until it can finally reach the goal. See Figure 1 (a) for a visual representation of the problem. A first heuristic one could devise is to encourage the car to gain height: $\Phi(s) = height(s)$. Since the goal location is at the top of a hill, this makes sense. But, as the car needs to build up momentum by driving up and down the two hills, in many situations, the car should actually choose to go down instead of trying (and failing) to get further up the hill. Thus, this rule is not perfect. Another heuristic that can be devised is to encourage increasing speed: $\Phi(s) = speed(s)$. This also makes sense, as the underpowered car needs some initial speed to climb up the hill. But, again, the car needs to drive up the hills many times, each time slowing down in the process, so this heuristic is not perfect either. Basically, the car is constantly trading potential and kinetic energy until it can reach the goal. While both heuristics could be useful for helping the agent solve the task, it is unclear how we could optimally combine them without going through an extensive tuning and engineering phase.

To overcome this problem, we propose an ensemble approach to reward shaping with multiple heuristics that automates the process of combining them.



(a)

(b)

Figure 1: (a) A visual representation of Mountain Car. (b) An ensemble of shapings in Mountain Car. The ensemble of shapings approximates the performance of the best shaping, which was unknown *a priori*.

3 Ensembles of Shapings

Ensemble techniques were developed to combine multiple 'weak' decision making algorithms into a stronger 'super' decision maker, aiming to outperform any of the constituting components. We propose to apply this idea to combine many 'weak' or suboptimal heuristics in reinforcement learning.

To do so, we first create multiple copies of the reward signal, each injected with a different potential-based reward shaping function. Given scalar reward function R and potential-based shaping functions F_1 through F_m, we construct a multi-objective reward function $\mathbf{R} = [R + F_1, \ldots, R + F_m]$ [13, 17, 18]. This process is called *multi-objectivization*. Each of the individual signals encodes a different piece of heuristic information and could be used on its own to solve the task, but we posit that creating an ensemble using these signals can help to solve the task faster by combining the different heuristics automatically.

An ensemble is then created by having m off-policy learning algorithms learn in parallel on the same experiences, each evaluating the behaviour according to one of the different enriched reward signals. The Horde [19] architecture is well-suited for this purpose given its off-policy convergence guarantees with linear function approximation and its computational efficiency, although it does place restrictions on the behaviour policy. In practice, an ensemble of Q-learners may work equally well, although it lacks convergence guarantees as strong as Horde.

An ensemble policy π is derived by combining each component's preferences:

$$\pi(s) = \arg\max_{a} \sum_{i}^{m} p_i(s, a)$$

These preferences could be simple votes or rankings [20] or more complex dynamic, confidence-based preferences [17]. The preferences of each ensemble component will be biased by the heuristic that component is using, and employing a combination mechanism like majority voting ensures that the ensemble action will be their common denominator. Thus, even though heuristics do not apply in every situation, they can compensate for each other's suboptimality.

4 Mountain Car

In the Mountain Car task, discussed above, a learning agent receives a negative reward for every step taken, and this sequence of negative rewards only stops when he arrives at the goal location. Therefore, the optimal policy, maximizing the accumulated rewards, reaches the goal in a minimum number of steps. The reward function itself is very uninformative, as it does not provide any gradient information towards the goal. We suggested using height and speed as heuristics to help the learning agent find such behaviour faster. A third heuristic we will investigate is encouraging the agent to move to the right, since the goal is located at the far right of the world.

In Figure 1, we compare the performance of the $GQ(\lambda)$ reinforcement learning algorithm [21] learning to solve Mountain Car without shaping, with a single one of the three proposed shaping functions, and with an ensemble of shapings encoded in the Horde architecture. Majority voting is employed as the mechanism to derive a policy from the ensemble. We observe that the speed shaping is the most useful of the three heuristics when used on their own, and, while this knowledge was not available *a priori* , the ensemble automatically approximates this best performance.

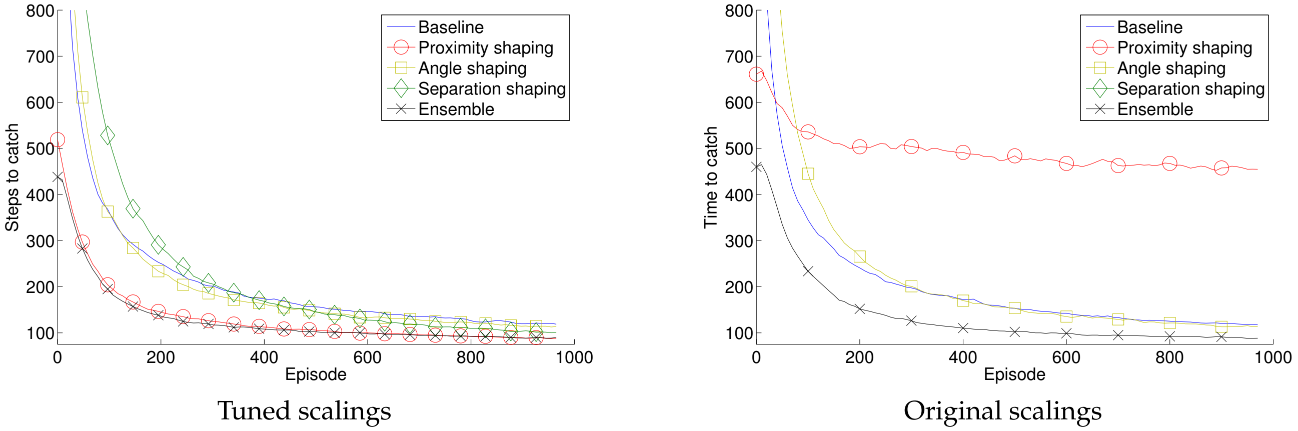


Figure 2: An ensemble of shapings in Predator-Prey. The scaling of the individual shapings has a large impact on performance, while the ensemble is unaffected.

5 Predator-Prey

We perform a similar experiment in the multi-agent Predator-Prey or Pursuit domain [22]. Two predators must learn how to coordinate in order to catch a fleeing prey, and only receive a positive reward when the prey is caught by at least one of the predators. Three heuristics we propose that could help are: (1) encouraging proximity to the prey, (2) encircling of the prey, and (3) separation between the predators. Figure 2 (a) compares the performance between standard $Q(\lambda)$-learning without shaping, with one of the three shapings, and an ensemble that uses a confidence measure to combine the three shapings [17]. The ensemble automatically, without prior knowledge, outperforms the best shaping alone (about 15% improvement in initial performance), even though two of the three shapings on their own are ineffective or detrimental to performance compared with the baseline.

An issue with the previous experiments is that we needed to tune the magnitudes of the shapings in order to provide the best performance. Of course, all this tuning is counterproductive, since our goal is to minimize the sample complexity of reinforcement algorithms, whereas tuning requires extra samples to select the best performing variants. Now, we argue that ensembles of shapings are not only robust with respect to the quality of the different heuristics provided, but also to their scalings. Figure 2 (b) shows the results of the same Predator-Prey experiment as in the (a) part of that figure, except that we simply left the magnitudes of the shapings as they were pre-tuning. In this case, every shaping on its own is detrimental to performance, yet the ensemble performs similar to the situation with the tuned scalings.

In other work, we have taken this one step further, by including multiple versions of the same heuristic in the ensemble, each version differently scaled [23]. In that paper, we present experiments in Mountain Car and Cart Pole that show how such an ensemble completely removes the need for tuning, automatically approximating the fastest possible learning given several shapings and arbitrary ranges of scalings. That is the first approach to reward shaping that truly removes the need for tuning.

6 Conclusions and Future Work

Reward shaping is a useful tool to incorporate prior knowledge to help a reinforcement learning agent reduce the number of experiences required to reach a desirable level of performance. Yet, in order for the shaping to be successful, there is usually some tuning necessary with respect to what knowledge to include, and how to scale the magnitude of the shaping compared to the base reward signal's magnitude. The number of extra experiences required for this tuning phase will typically be much higher than what is gained in the end by applying the best shaping.

In this work, we have investigated the use of shaping ensembles to remove this need for tuning, proposing an off-the-shelf solution that automatically can combine many different pieces of information. Ideally, a system designer can now create a number of shaping functions based on the information he has (even creating multiple shapings with the same piece of knowledge, but differently scaled) and combine them in an ensemble that automatically uses this information in a good, if not best, way.

As we discussed before, reward shaping can be used to encode things other than heuristic expert knowledge. Therefore, ensembles could for example be used to achieve multi-task transfer, assuming each source task will contribute different information to the target task, or to incorporate demonstrations given by different experts, assuming different experts' demonstrations are significantly different. The ultimate goal is to demonstrate how an ensemble provided with a number of these types of information can allow a reinforcement learning agent to solve a complex practical application, such as a robotics manipulation task.

References

[1] B. R. Leffler, M. L. Littman, and T. Edmunds, "Efficient reinforcement learning with relocatable action models," in *AAAI*, vol. 7, pp. 572–577, 2007.

[2] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning*, vol. 99, pp. 278–287, 1999.

[3] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *The Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

[4] T. Brys, A. Harutyunyan, M. E. Taylor, and A. Nowé, "Policy transfer using reward shaping," in *International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2015.

[5] M. E. Taylor, H. B. Suay, and S. Chernova, "Integrating reinforcement learning with human demonstrations of varying ability," in *The 10th International Conference on Autonomous Agents and Multiagent Systems*, pp. 617–624, 2011.

[6] T. Brys, A. Harutyunyan, M. E. Taylor, and A. Nowé, "Reinforcement learning from demonstration through shaping," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[7] W. B. Knox and P. Stone, "Combining manual feedback with subsequent mdp reward signals for reinforcement learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 5–12, 2010.

[8] R. Loftin, J. MacGlashan, B. Peng, M. E. Taylor, M. L. Littman, J. Huang, and D. L. Roberts, "A strategy-aware technique for learning behaviors from discrete human feedback," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-2014)*, 2014.

[9] A. Harutyunyan, T. Brys, P. Vrancx, and A. Nowé, "Shaping mario with human advice," in *International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2015.

[10] M. Grześ and D. Kudenko, "Online learning of shaping rewards in reinforcement learning," *Neural Networks*, vol. 23, no. 4, pp. 541–550, 2010.

[11] S. Devlin, D. Kudenko, and M. Grześ, "An empirical study of potential-based reward shaping and advice in complex, multi-agent systems," *Advances in Complex Systems*, vol. 14, no. 02, pp. 251–278, 2011.

[12] K. Efthymiadis and D. Kudenko, "Using plan-based reward shaping to learn strategies in Starcraft: Broodwar," in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, pp. 1–8, IEEE, 2013.

[13] T. Brys, A. Harutyunyan, P. Vrancx, M. E. Taylor, D. Kudenko, and A. Nowé, "Multi-objectivization of reinforcement learning problems by reward shaping," in *International Joint Conference on Neural Networks (IJCNN)*, pp. 2315–2322, IEEE, 2014.

[14] H. Kim, M. I. Jordan, S. Sastry, and A. Y. Ng, "Autonomous helicopter flight via reinforcement learning," in *Advances in neural information processing systems*, 2003.

[15] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, "Autonomous inverted helicopter flight via reinforcement learning," in *Experimental Robotics IX*, pp. 363–372, Springer, 2006.

[16] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine learning*, vol. 22, no. 1-3, pp. 123–158, 1996.

[17] T. Brys, A. Nowé, D. Kudenko, and M. E. Taylor, "Combining multiple correlated reward and shaping signals by measuring confidence," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1687–1693, 2014.

[18] A. Harutyunyan, T. Brys, P. Vrancx, and A. Nowé, "Off-policy shaping ensembles in reinforcement learning," in *European Conference on Artificial Intelligence*, 2014.

[19] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup, "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 761–768, International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[20] M. A. Wiering and H. van Hasselt, "Ensemble algorithms in reinforcement learning," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 4, pp. 930–936, 2008.

[21] H. R. Maei and R. S. Sutton, "Gq (λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces," in *Proceedings of the Third Conference on Artificial General Intelligence*, vol. 1, pp. 91–96, 2010.

[22] M. Benda, V. Jagannathan, and R. Dodhiawala, "On optimal cooperation of knowledge sources - an empirical investigation," Tech. Rep. BCS–G2010–28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, WA, USA, July 1986.

[23] A. Harutyunyan, T. Brys, P. Vrancx, and A. Nowé, "Multi-scale reward shaping via an off-policy ensemble," in *International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2015.