End-Effect Exploration Drive for Effective Motor Learning

No Author Given

No Institute Given

Abstract. Stemming on the idea that a key objective in reinforcement learning is to invert a target distribution of effects, end-effect drives are proposed as an effective way to implement goal-directed motor learning, in the absence of an explicit forward model. An end-effect model relies on a simple statistical recording of the effect of the current policy, here used as a substitute for the more resource-demanding forward models. When combined with a reward structure, it forms the core of a lightweight variational free energy minimization setup. The main difficulty lies in the maintenance of this simplified effect model together with the online update of the policy. When the prior target distribution is uniform, it provides a ways to learn an efficient exploration policy, consistently with the intrinsic curiosity principles. When combined with an extrinsic reward, our approach is finally shown to provide a faster training than traditional off-policy techniques.

Keywords: Reinforcement Learning · Intrinsic reward · Model-based Exploration · Motor learning

1 Introduction

Recent developments in artificial intelligence have produced important qualitative leaps in the field of pattern recognition, in video games and assisted driving. However, a number of tasks considered as simple, are struggling to find a convincing artificial implementation... This is the field of action selection and motor control. For instance, the fine manipulation of objects, as well as movement in natural environments, and their combination through real time motor control, remain major scientific challenges at present. Compared to the case of video games, reinforcement learning, for instance, remains rather limited in the field of robotic and motor control. The huge improvements "from-scratch" obtained in virtual environments are difficult to transfer to real robotics, where millions of plays can not be engaged under risk of breakage, and simulators are expensive to develop. The brain capability to develop motor skills in a very wide range of areas has thus no equivalent in the field of artificial learning.

The control of a dynamic system generally relies on a *model* [12]. When a certain objective is given, it can act so as to reach the objective through *model inversion*. This kind of controller needs a forward model, that can be given, or trained from the sensory data. A main drawback of model inversion is the fact that, in most cases of interest (like the control of an articulated body), the models are not invertible and no single definite control can be established from a given objective [6].

The counterpart to analog control is digital (discrete) control. In order to learn how to act, the agent is guided by a reward signal, that is much cheaper to extract than an exact set point. Then the choice of the action is monitored by a scalar quantity named the utility, that is the long term sum of rewards [17]. Supported by the solid principles of dynamic programming, this agent is in principle easier to train than an analog one, but a hidden difficulty lies in finding a good training dataset for the agent. A classical problem in that case is the lack of sampling efficacy due to the sparsity of the rewards, or to the closed-loop structure of the control task, where the examples encountered are statistically dependent on the controller parameters, providing a risk of a self referential loop and local optimum.

A main difference between the two approaches is thus the presence (or the absence) of a definite model of the external world. Knowing the effect of our own actions in the world provides ways to anticipate and do planning to reach an objective, at the cost of maintaining and updating a model of the world [9]. The trade-off between model-free and model-based control has provoked many debates in the reinforcement learning community, with a preference toward model-free approaches for they are cheaper to maintain and easier to control, leaving unsolved the problem of the sampling efficacy and causing very long training sessions in moderately complex environments.

Our argument here is that the sampling efficacy is bound to the problem of training a model, and one can not expect to have an efficient sampling without a minimal model of the *effect* of action. This model does not need to be perfectly accurate, but it needs to be good enough to allow the agent to efficiently sample its environment in order to grab all the disposable information in relation to the task at hand. We assert here that a simple *effect model* is enough to provide all the needed variability in probing the effect of an action or a policy.

2   Method

2.1   A probabilistic view to motor supervision

The probabilistic view to learning cause-effect relationships is at the core of many recent developments in machine learning, with a body of optimization techniques known as "variational inference" implementing model training from data [8]. We assume for simplicity that the environment is not hidden to the agent, i.e. the environment is fully observable. We also assume a discrete updating of states and actions, like in classic reinforcement learning. Then, if s is the state of the environment (or a context), and a an action performed by the agent, consider e as the *effect* of the action performed by the agent in that particular state. The effect can be a short-term effect, like reading a new state from the environment. It can also be a long term effect, like winning or loosing a game, or reaching an objective s^* in the future. Because there is a lot of uncertainty on the effect of an action, it is modeled as a probability distribution $p(E|s,a)$. When the effect is not dependent on a context, it can be noted more simply $p(E|a)$. Given a certain policy $\pi(a|s)$, one can also consider the average distribution of effects obtained when applying that specific policy, namely

$$p(E|s) = \mathbb{E}_{a \sim \pi(A|s)} p(E|s,a) \tag{1}$$

In goal-directed control, if e is an expected effect, an *inverse control policy*, whose role is to maximize the chance to reach the effect, can be defined using Bayes rule as:

$$\pi(a|s,e) = \frac{p(e|s,a)\pi(a|s)}{p(e|s)} \tag{2}$$

That is the inversion of the model in a probabilistic setup [1].

Assume now $p^*(e|s)$ be a target distribution of effects. This distribution is distinct from $p(e|s)$ that is the distribution of effects under the current policy. It is assumed to be realizable from an (unknown) target policy $\pi^*(a|s)$, that can be decomposed into:

$$\pi^*(a|s) = \mathbb{E}_{e \sim p(E|s,a)} \pi^*(a|s,e) \frac{p^*(e|s)}{p(e|s,a)} \tag{3}$$

The right side of the equation provides an estimation of the optimal policy based on a sample e of the effect of the action. Unfortunately, the optimal inverse control policy $\pi^*(a|s,e)$ is unknown. A shortcut is to approximate it with the current inverse control policy $\pi(a|s,e)$. In that case, it happens from equation (2) that the formula simplifies to :

$$\pi^*(a|s) \simeq \mathbb{E}_{e \sim p(E|s,a)} \pi(a|s,e) \frac{p^*(e|s)}{p(e|s,a)} \tag{4}$$

$$= \pi(a|s) \mathbb{E}_{e \sim p(E|s,a)} \frac{p^*(e|s)}{p(e|s)} \tag{5}$$

This formula shows that a correction term can be applied to the current policy without owning an explicit forward model, but rather through a mere *sampling* of the effect of the action. This forms the basis of a supervised approach to motor learning, allowing to update a policy so as to reach a target distribution.

For instance, in a dicrete setup, assuming there exists a $Z(s) \in \mathbb{R}$ such that $\log \pi(a|s) = \beta Q(s,a) + Z(s)$ (softmax policy) makes it possible to update $Q(s,a)$ with the last effect sample e like:

$$Q(s,a) \leftarrow Q(s,a) - \frac{\alpha}{\beta}(\log p(e|s) - \log p^*(e|s)) \tag{6}$$

This update renders the current policy closer to the optimal one. A side effect of this update is that it also changes the effect model that includes the contribution of the policy (see eq. (1)). Repeating the operation with a small α and different samples of a and e should, on average, reduce the divergence between π and π^*. Equation (6) also provides an interesting identity when considering the classical (reward-based) TD-error, with $\frac{1}{\beta}(\log p(e|s) - \log p^*(e|s))$ taking the role of the TD error, i.e. being identified with $Q(s,a) - \tilde{R}(e)$ (with $\tilde{R}(e)$ a putative sum of rewards up to e), making it possible, for instance, to set up an intrinsic reward implementing a policy that realizes a known prior on the effects. This intrinsic reward is called here the "End-Effect Drive".

This supervised approach to policy relies on an effect model $p(e|s)$ that is less detailed than a forward model. Various kinds of approximate forward models can be found in goal-directed motor control literature, like dynamic goals [7] and distal teacher [6], though generally learning to associate the current action with a distal effect [13,9]. In our case, no specific action-effect association is provided by the model and only the sampling of the environment dynamics, whatever the action, is considered. This "action-agnostic" is close to the concept of state-visit counter, as it is proposed in [2].

2.2   A uniform exploration drive

An important special case is when the objective is not to reach a certain effect, but rather to explore uniformly the range of all possible effects. In that case, the objective effect distribution p^* is uniform over the effect space. This kind of supervision can be seen as a generalization of multiple-goal supervision [5] toward defining each possible effect as a goal. The expected outcome of this uniform drive is to provide a uniform sampling over the effect space, i.e. implement a uniform exploration policy. This intrinsic reward is called here the "End-Effect Exploration Drive" (E3D in short). It is consistent with the pseudo-count bonus proposed in [2]. A similar drive was also proposed in a recent draft as the "state marginal matching" drive [10].

By construction, the E3D is positive when e is rarely visited under the current policy ($p(e|s) < p^*(e)$), and negative the other way. It thus tries to promote rare and "surprising" effects, and lower the occurrence of habitual "boring" effects. It must be noticed that the promotion of rare effects tends to make them less rare, and the rejection of habitual effects tends to make them less habitual, up to an equilibrium where the log-probability ratio should be close to zero. The term "reward" is problematic in that case because the reward obtained is dependent on the policy, and a policy that follows the E3D drive will tend to get less and less rewards. This is at odd with the principle of policy optimization through reward maximization, and needs to be counterbalanced with some form of utility drive. This is the reason why a variational inference setup is particularly well suited in that case, with the distal uniform drive taking the role of a prior under a variational formulation.

2.3   Link with variational inference

A key intuition in Friston's "unified brain theory" paper [4] is the mapping of the utility, as it is defined in economy and reinforcement learning, with a measure of negative surprise (i.e the log probability over the sensory data distribution). Combined with a prior distribution in the control space, the action takes the role of a latent variable that is updated so as to reduce the prediction error with regards to the prior, much like in predictive coding.

The unified approach proposed by Friston and colleagues is more generally consistent with the variational auto-encoder principles, in which a latent description of the data is constructed so as to implement a trade-off between the complexity of the description and the accuracy of the prediction. Variational reinforcement learning was recently proposed as a way to reconcile the discrete view to motor control with the continuous processing of the latent variable in variational auto-encoders [11,3,5], with the motor command playing the role of a latent code for the reward data. In our simplified writing, the utility maximization (or surprise minimization) rests on minimizing:

$$-\mathbb{E}_{a\sim\pi(a|s);e\sim p(e|s,a)}\beta R(e) + \mathrm{KL}(\pi(a|s)||\pi^*(a)) \tag{7}$$

with $R(e)$ here a measure of the sum of (extrinsic) rewards, up to e. Interestingly, the current policy $\pi(a|s)$ lies at the crossroad of a reference (maximum entropy) policy π^* and reward maximization, with the softmax policy representing a compromise between both tendencies in the discrete case.

Extending toward a uniform prior over the space of effects can be written when both considering the motor command and the effect as latent variables that may both explain the current observation, that writes:

$$-\mathbb{E}_{a,e\sim p(a,e|s))}\beta R(e) + \mathrm{KL}(p(a,e|s))||p^*(a,e)) \tag{8}$$

For the purpose of illustration, we propose here an additional simplification, that is assuming an *independence* of both factors (e and a) on causing the current data (Naïve Bayes assumption), dividing the KL term in two parts:

$$-\mathbb{E}_{a\sim\pi(a|s);e\sim p(e|s,a)}\beta R(e) + \mathrm{KL}(\pi(a|s)||\pi^*(a)) + \mathrm{KL}(p(e|s)||p^*(e)) \tag{9}$$

This forms the baseline of our variational policy update setup. The optimization, that is done on $\pi(a|s)$, obeys in that case on a triple constraint, that is maximizing the reward through minimizing both the distance to a baseline policy and the distance of the effect to a reference (supposedly uniform) effect distribution.

In a discrete setup, the uniform prior on the action is supposed implemented with the softmax decision rule. It is then sufficient to assume the following update for the action-value function. After reading e, the TD-error should be defined as:

$$\mathrm{TD}(s,a,e) = \lambda(Q(s,a) - R(e)) + \frac{1}{\beta}(\log p(e|s) - \log p^*(e))) \tag{10}$$
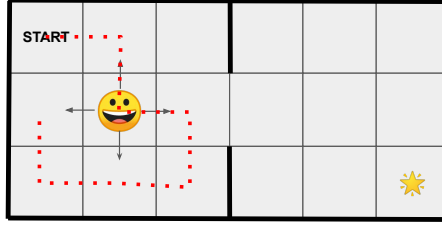
Fig. 1: A simple two-rooms environment. Starting from the upper-left corner, the agent is asked to plan a full sequence made of 7 elementary actions $a_1, ..., a_7$, each elementary action being in (E,S,W,N). The only read-out from the environment is the final state, and a reward, that is equal to 1 if the final state is the lower-right corner, and 0 otherwise.

With λ a precision hyperparameter accounting for the different magnitudes of rewards, allowing to manipulate the balance between reward seeking and exploration-seeking drives. Interestingly, the reward $R(e)$ has here the role of a regularizer with regards to the current Q-value. The sum of the future rewards can be estimated using the classical Bellman recurrence equation, i.e. $Q(s, a) \sim r(s, a) + Q(s', a')$, in which case the training procedure needs to maintain an estimate of a standard action-value function $Q_{\text{ref}}(s, a)$ to update the actual parameters of the policy $Q(s, a)$.

3   Results

We present in simulation a pilot implementation of the principle presented in the previous section. The principal idea is to illustrate an important feature of biological motor control, that is the control of an effector showing many degrees of freedom, like e.g. an articulated limb with many joints.

Let $\mathcal{A}$ a control space accounting for a single degree fo freedom (here a discrete set of actions i.e. $\mathcal{A} = \{E, S, W, N\}$), each motor command owning n degrees of freedom, i.e. $a_{1:n} = a_1, ..., a_n \in \mathcal{A}^n$. The effect space is expected to be much smaller, like it is the case in end-effector control, where only the final set point of a movement in the peripheral space is considered as the result of the action. Each degree of freedom is supposed to be independent, i.e. the choice of a_i does not depend on the choice of a_j, so that $\pi(a_{1:n}|s_0) = \pi(a_1|s_0) \times ... \times \pi(a_n|s_0)$. When a single context s_0 is considered, the policy writes simply $\pi(a_{1:n})$. The size of the action space is thus combinatorially high, and one can not expect to enumerate every possible action in reasonable computational time. In contrast, the effect space is bounded, and the number of all final states can be enumerated. However, the environment is constructed in such a way that some final states are very unlikely to be reached under a uniform sampling of the action space.

The environment we consider is a grid world with only 18 states and two rooms, with a single transition allowing to pass from room A to room B (see figure 1). Starting in the upper left corner of room A, the agent samples a trajectory $a_{1:7} \in \mathcal{A}$ from a policy π, that trajectory being composed of 7 elementary displacements. The agent does not have the capability to read the intermediate states it is passing through, it can only read the final state after the full trajectory is realized. In such an environment, a baseline uniform exploration does not provide a uniform distribution of the final states. In particular, when acting at random, the chance to end-up in the first room is significantly higher than the chance to end up in the second room.

The agent starts from scratch, and has to build a policy $\pi(a)$ and an effect model $p(s_n)$, with s_n the final state. There are two task at hand. A first task is a simple exploration task and the objective is to uniformly sample the effect space, which should imply a non-uniform sampling policy. A second task consists in reaching the lower-right corner, the state that shows the lowest probability with a uniform sampling. For that, a reward of 1 is given when the agent reaches the lower corner, and 0 otherwise.

The update procedure is made of a single loop (see algorithm 1). The update is done online at each trial. Both the policy and the effect model are updated, with different training parameters. The general idea is to train the effect model a little more "slowly" than the policy, for the policy improvement to firmly take place before they are passed on the effect model.

Stemming from a uniform policy, the effect of the E3D drive is to render the "rare" states more attractive, for they are bound with a positive intrinsic reward, while the more commonly visited states are bound with a negative reward, that reflects a form of "boredom". Marking a rare state as "attractive" tends to increase the number of visits, and finally lower the initial positive reward. In the case of a "gradient" in the likelihood of the final states, with a number of final visits inversely proportional to the distance to the initial state,
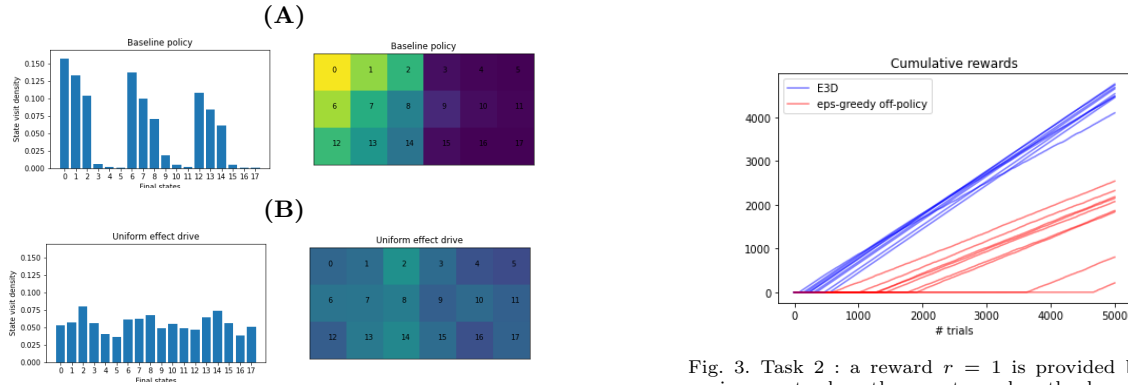
Algorithm 1 End-Effect Exploration Drive (E3D)

Require: α, β, λ, η

   $Q \leftarrow \mathbf{0}_{|\mathcal{A}| \times n}$
   $p \leftarrow$ Uniform
   $p^* \leftarrow$ Uniform
   while number of trials not exceeded **do**
      sample $a_{1:n} \sim \pi(A_{1:n})$
      read s_n, r
      $p \leftarrow (1 - \eta)p + \eta \mathbf{1}_{S = s_n}$
      for $i \in 1..n$ **do**
         $Q(a_i) \leftarrow (1 - \alpha\lambda)Q(a_i) + \alpha\lambda r - \frac{\alpha}{\beta}(\log p(s_n) - \log p^*(s_n))$
      end for
   end while

(A)

(B)

Fig. 2. Task 1 : no reward is provided by the environment. Empirical distribution of the final states, after 5000 trials. **A.** Uniform policy. **B.** End-Effect Exploration Drive (E3D) algorithm. $\alpha = 0.3$, $\beta = 1$, $\lambda = 0.03$, $\eta = 10^{-2}$.

Fig. 3. Task 2 : a reward $r = 1$ is provided by the environment when the agent reaches the lower-right corner. Cumulative sum of rewards over 5000 trials, on 10 training sessions. The E3D algorithm is compared with state-of-the-art epsilon-greedy update. $\alpha = 0.3$, $\beta = 100$, $\lambda = 0.03$, $\eta = 10^{-2}$, $\varepsilon = 0.1$.

the E3D drive favors a progressive "expansion" of the visiting territory, for each peripheral state attained will increase the probability to reach its further neighbors, up to the final limit of the state space. In small environment like the one proposed here, the limit is rapidly attained and a rapid alternation of visits is observed over the full state space.

The final distribution of states is compared in figure 2 in the case of a uniform policy and the E3D drive. In that specific setup, a strong bias in favor of the first room is observed, and a gradient of likelihood is observed from the initial state toward the lower right corner (figure 2A). In contrast, a time consistent uniform pattern of visit is observed in the second case, that illustrates the capability of the E3D drive to set up specific polices devoted to the wide exploration of the environment.

When a reward r is provided by the environment, the question comes whether to balance the policy update procedure in favor of seeking for rewards or seeking for novel effects. By construction, the exploration drive is insensitive to the value of β, for the update is exactly proportional to $\frac{1}{\beta}$. A high β is associate with a small update and vice versa. this is not the case for the leftmost part of the update (equation 10). A high β render the agent more sensitive to the extrinsic rewards. In practice, while no reward (or a uniform reward) is provided by the environment, the agent is only guided by the exploration drive. Once a reward is encountered, it tends to overtake the initial uniform exploration, providing a firm tendency toward a reward-effective selection of action. This is in contrast with the standard epsilon-greedy strategy, imposing to balance the exploration/exploitation trade-off by hand.

The E3D approach finally provides an Online/on-Policy training procedure that conforms to the main requirements of efficient reinforcement learning, showing both an efficient exploration policy when the rewards are sparse, and the capability to monitor the exploration/exploitation tradeoff with the inverse temperature β in function of the magnitude of the rewards.

The cumulative rewards obtained with the E3D update and a state-of-the art off-policy/epsilon greedy update are compared in figure 3, with ε set to 0.1. If both techniques manage to reach a reward-efficient policy in the long run, the exploration strategy developed in E3D makes it easier to reach the rewarding state, providing the reward earlier in time and developing a fast-paced reward-seeking strategy that overtakes the baseline approaches.

4   Conclusions

Despite its simplicity, our pilot training setup allows to illustrate the main features expected from the inversion of a target distribution of effects, that is the capability to rapidly explore the environment through a reciprocal update of the policy and the effect model. We found in practice that the update of the model needs to be a bit slower than that of the policy to allow for the policy to improve over time and increase the extent of the effect space in a step by step manner. By balancing the effect of the rewards with the inverse temperature parameter, it is possible to catch and exploit very sparse rewards in large environments.

The model is developed here as a first draft in an "end-effect" setup, with very little influence of the context or the states visited in the monitoring of the policy. Extensions toward closed-loop state-action policies is not far from reach, for equation (10) allows to exploit the Bellman recurrence to guide the exploration-driven policy with a reference action-value function, that should be updated in parallel to the model and the current policy. Extensions toward continuous action spaces are also needed in order to address effective motor control learning, which resorts to a deeper interpretation of our approach toward the variational inference setup.

References

1. Bays, P.M., Wolpert, D.M.: Computational principles of sensorimotor control that minimize uncertainty and variability. The Journal of physiology **578**(2), 387–396 (2007)
2. Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., Munos, R.: Unifying count-based exploration and intrinsic motivation. In: Advances in neural information processing systems. pp. 1471–1479 (2016)
3. Fox, R., Pakman, A., Tishby, N.: Taming the noise in reinforcement learning via soft updates. arXiv preprint arXiv:1512.08562 (2015)
4. Friston, K.: The free-energy principle: a unified brain theory? Nature reviews neuroscience **11**(2), 127–138 (2010)
5. Haarnoja, T., Tang, H., Abbeel, P., Levine, S.: Reinforcement learning with deep energy-based policies. arXiv preprint arXiv:1702.08165 (2017)
6. Jordan, M.I., Rumelhart, D.E.: Forward models: Supervised learning with a distal teacher. Cognitive science **16**(3), 307–354 (1992)
7. Kaelbling, L.P.: Learning to achieve goals. In: IJCAI. pp. 1094–1099. Citeseer (1993)
8. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
9. Kurutach, T., Clavera, I., Duan, Y., Tamar, A., Abbeel, P.: Model-ensemble trust-region policy optimization. arXiv preprint arXiv:1802.10592 (2018)
10. Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., Salakhutdinov, R.: Efficient exploration via state marginal matching. arXiv preprint arXiv:1906.05274 (2019)
11. Levine, S., Koltun, V.: Guided policy search. In: International Conference on Machine Learning. pp. 1–9 (2013)
12. Miall, R.C., Wolpert, D.M.: Forward models for physiological motor control. Neural networks **9**(8), 1265–1279 (1996)
13. Mishra, N., Abbeel, P., Mordatch, I.: Prediction and control with temporal segment models. arXiv preprint arXiv:1703.04070 (2017)
14. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
15. Mohamed, S., Rezende, D.J.: Variational information maximisation for intrinsically motivated reinforcement learning. In: Advances in neural information processing systems. pp. 2125–2133 (2015)
16. Schmidhuber, J.: Curious model-building control systems. In: Proc. international joint conference on neural networks. pp. 1458–1463 (1991)
17. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
18. Toussaint, M.: Robot trajectory optimization using approximate inference. In: Proceedings of the 26th annual international conference on machine learning. pp. 1049–1056 (2009)