Solving deterministic policy (PO)MDPs using Expectation-Maximisation and Antifreeze

Thomas Furmston and David Barber

Department of Computer Science
University College London
London WC1E 6BT, UK

Abstract. The viewpoint of solving Markov Decision Processes and their partially observable extension refers to finding policies that maximise the expected reward. We follow the rephrasing of this problem as learning in a related probabilistic model. Our trans-dimensional distribution formulation obtains equivalent results to previous work in the infinite horizon case and also rigorously handles the finite horizon case without discounting. In contrast to previous expositions, our framework elides auxiliary variables, simplifying the algorithm development. For any MDP the optimal policy is deterministic, meaning that this important case needs to be dealt with explicitly. Whilst this case has been discussed by previous authors, their treatment has not been formally equivalent to an EM algorithm, but rather based on a fixed point iteration analogous to policy iteration. In contrast we derive a true EM approach for this case and show that this has a significantly faster convergence rate than non-deterministic EM. Our approach extends naturally to the POMDP case as well. In the special case of deterministic environments, standard EM algorithms break down and we show how this can be addressed using a convex combination of the original deterministic environment and a fictitious stochastic 'antifreeze' environment.

1 Markov Decision Processes

A Markov decision process (MDP) is defined on state-variables $x_t = 1, \ldots, X$, actions $a_t = 1, \ldots, A$, and utilities (rewards) u_t, at times $t = 1, \ldots, T$. The model describes situations in which an agent is in state $x_t = \mathsf{x}$, decides to take action $a_t = \mathsf{a}$, and receives a utility $u_t(x_t = \mathsf{x}, a_t = \mathsf{a}) = \mathsf{u}$ from the environment. The system is assumed to be Markovian such that a state-action trajectory can be described by the distribution

$$p(x_{1:t}, a_{1:t}|\pi) = p(x_1)p(a_1|x_1, \pi) \prod_{\tau=1}^{t-1} p(x_{\tau+1}|x_\tau, a_\tau)p(a_\tau|x_\tau, \pi) \tag{1}$$

where the environment is described by the transition $p(x_{\tau+1}|x_\tau, a_\tau)$ and the policy $p(a_\tau|x_\tau, \pi)$. For simplicity we assume that the transitions and policy distributions are stationary, with the extension to the non-stationary case being

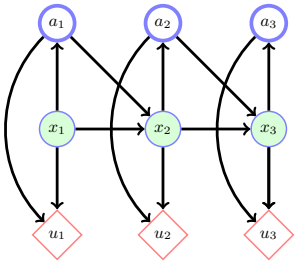


Fig. 1. A MDP represented as an Influence Diagram. We define utilities (rewards) to depend on the current state and action, $u_t(x_t, a_t)$. The policy $p(a_t|x_t)$ determines the decision and the environment is modeled by the transition $p(x_t|x_{t-1}, a_{t-1})$. The diagram represents $\sum_t p(x_t, a_t)u_t(x_t, a_t)$ where the marginal $p(x_t, a_t)$ is determined from the Markov chain, equation (1).

straightforward. Given a transition and policy, the Markov chain has an occupancy probability $p(x_t, a_t)$ computed by marginalizing equation (1). Associated with each state x_t and action a_t there is a utility $u_t(x_t, a_t)$. The total expected utility over an horizon T is then given by

$$u(\pi) = \sum_{t=1}^{T} \sum_{x_t, a_t} u_t(x_t, a_t) p(x_t, a_t|\pi) \tag{2}$$

It is common to constrain the non-stationary utility to be a discounted stationary utility using a factor $\gamma \in [0, 1]$, so that $u_t(x_t x_t) \equiv \gamma^t u(x_t, a_t)$ to encourage the agent to take actions that maximise utility quickly. In the case of infinite horizons, the discount factor $\gamma < 1$ makes $u(\pi)$ finite. The setup may be depicted using an Influence Diagram[4], see figure 1. The MDP task is to find the policy π^* that maximizes (2). This is a standard problem in operational research, control and reinforcement learning, for which the classical algorithms are based on variants of policy and value iteration[8]. As an alternative to these classical procedures we cast this as a probabilistic inference problem allowing us to use various methods from that field. In particular we will find π^* through an Expectation-Maximisation (EM) style-algorithm[2]. Our work builds on the general approach introduced in [1] and the viewpoint of solving MDPs as likelihood optimization in a corresponding probabilistic model[9, 6].

2 Solving MDPs using EM

Our construction of a probabilistic framework in the case of an MDP is similar to that in [9] but doesn't require the introduction of auxiliary variables or a time prior. In order to maximise the total expected utility (2) we first construct a lower bound on (2). Without loss of generality we consider non-negative utilities which enables us to form a distribution $\hat{p}(x_{1:t}, a_{1:t}, t|\pi)$, whose normalization constant is equal to (2). This is achieved by defining the trans-dimensional distribution

$$\hat{p}(x_{1:t}, a_{1:t}, t|\pi) \equiv \frac{u_t(x_t, a_t)p(x_{1:t}, a_{1:t}|\pi)}{u(\pi)} \tag{3}$$

The reader may verify that this is a correctly normalised distribution since first summing over $x_{1:t}, a_{1:t}$ gives the expected utility at time t in the numerator

term. Then summing over t from 1 to T gives the total expected utility $u(\pi)$ on the numerator, equalling the denominator term. In order to obtain a lower bound on (2) we now introduce an auxiliary distribution $q(x_{1:t}, a_{1:t}, t)$, which we call the q-distribution. Taking the Kullback-Leibler divergence between q and $\hat{p}$ gives[1]

$$KL(q||\hat{p}) = H(q) - \langle \log u_t(x_t, a_t) \rangle_q - \langle \log p(x_{1:t}, a_{1:t}|\pi) \rangle_q + \log u(\pi)$$

where $\langle \cdot \rangle_q$ denotes the average w.r.t. the q-distribution, and H is the entropy function. We now use the fact that the Kullback-Leibler divergence is non-negative $\forall$ distributions $\hat{p}$, q, to obtain the bound

$$\log u(\pi) \geq -H(q) + \langle \log u(x_t, a_t) \rangle_q + \langle \log p(x_{1:t}, a_{1:t}|\pi) \rangle_q \tag{4}$$

Instead of maximising equation (2) directly we may alternatively find π^* by maximising this bound. This corresponds to a form of EM algorithm that iteratively optimises this bound using the following two-step procedure:

E-step For fixed π^{old} find the best q that maximises the r.h.s of (4). For no constraint on q, this gives $q = \hat{p}(x_{1:T}, a_{1:T}, t|\pi^{old})$.

M-step For fixed q find the best π that maximises the r.h.s of (4). This is equivalent to maximising the 'energy' $\langle \log p(x_{1:t}, a_{1:t}|\pi) \rangle_q$ with respect to π.

2.1 E-step

If we place no functional restriction on the q-distribution we have that the maximum occurs when $KL(q||\hat{p}) = 0$, that is when $q(x_{1:t}, a_{1:t}, t) = \hat{p}(x_{1:t}, a_{1:t}, t|\pi^{old})$, where π^{old} is the policy of the previous M-step. The E-step consists of calculating the quantities required to perform the M-step, namely the marginals $q(x_\tau, a_\tau, t)$. This is straightforward since, as a graphical model, q is simply a chain distribution for which marginal inference can be achieved in linear time $O(T)$ via standard message-passing techniques[10, 9]. For completeness, we outline here a suitable dynamic programming method. As the q-distribution is a chain distribution it can be written simply in terms of the forward and backward messages as

$$q(x_\tau, a_\tau, t) \propto \pi^{old}_{a,x} \gamma^t \beta_{t-\tau}(x, a) \alpha_\tau(x)$$

where

$$\beta_{t-\tau}(x, a) = \begin{cases} u(a, x) & \text{if } t - \tau = 0 \\ \sum_{x', a'} \beta_{t+1-\tau}(x', a') p(x'|x, a) \pi_{x,a} & \text{otherwise} \end{cases} \tag{5}$$

$$\alpha_\tau(x) = \begin{cases} p_0(x) & \text{if } \tau = 1 \\ \sum_{x', a'} p(x|x', a') \pi_{x', a'} \alpha_{\tau-1}(x') & \text{otherwise} \end{cases} \tag{6}$$

[1] The summation over t is implicit in the average over $q(x_{1:t}, a_{1:t}, t)$.

2.2 M-step

In the M-step we are interested in maximising the distribution w.r.t π, so separating out the policy terms from (4) we obtain the energy term

$$E(\pi) = \sum_{t=1}^{T} \sum_{\tau=1}^{t} \sum_{x_\tau, a_\tau} q(x_\tau, a_\tau, t) \log \pi_{a_\tau, x_\tau} \tag{7}$$

where we use a tabular policy $\pi_{a_\tau, x_\tau} \equiv p(a_\tau | x_\tau, \pi)$. For a stationary policy the resulting M-step may be written as

$$\pi_{a,x} \propto \pi_{a,x}^{old} \sum_{t=1}^{T} \sum_{\tau=1}^{t} \gamma^t \beta_{t-\tau}(x, a) \alpha_\tau(x) \tag{8}$$

2.3 Relation to other EM algorithms

There are several other constructions of EM algorithms that have been designed to solve MDPs, for example [9, 1, 3]. In [9] the original MDP is cast as an infinite mixture model of finite-time MDPs. Each finite-time MDP is similar to the original MDP in that it has the same initial state, transition and policy probabilities, but differs in that it emits a binary signal, R, at the final time-step. The mixture weight is given by $p(t) = \gamma^t(1-\gamma)$. The auxiliary binary variable is introduced only for the purposes of constructing the EM algorithm, and is defined as

$$p(R = 1 | a_t = a, x_t = x) \propto u(a_t, x_t) \tag{9}$$

The mixture model of finite time MDPs is then given by

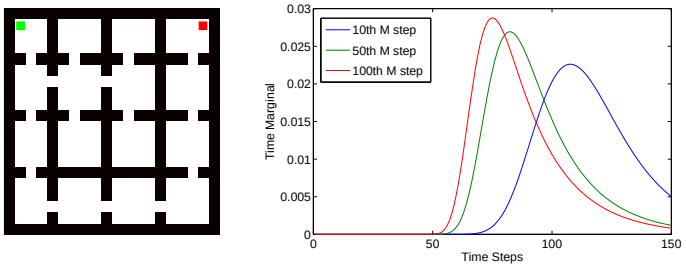
$$p(R, x_{1:t}, a_{1:t}, t; \pi) = p(R, x_{1:t}, a_{1:t} | t; \pi) p(t) \tag{10}$$

With this construction, maximising the likelihood, $p(R = 1; \pi)$, is equivalent to solving the original MDP. This likelihood maximization problem is solved through an EM algorithm which gives policy updates of the form

$$\pi_{a,x}^{new} \propto \pi_{a,x}^{old} \sum_{\tau=0}^{\infty} p(t = \tau) \beta_\tau(a, x) \tag{11}$$

Taking $T = \infty$ in (8) and performing some simple manipulations then one indeed obtains updates of the form (11).

Although our construction is similar to [9] there are several differences. In our construction it is unnecessary to introduce the auxiliary variable R, which we feel makes the construction clearer. It is also unnecessary for us to introduce a time prior $p(t)$, so that dealing with the case $\gamma = 1$ requires no special treatment. This means that our derivation can deal with the finite and infinite horizon cases in the same derivation.



(a) (b)

Fig. 2. (a) Maze considered in the MDP experiments. The walls are black, with initial state in the top left corner (green), the goal state in the top right corner (red) and the rest of the maze in white. There are in total 240 states. (b) The time marginal $\hat{p}(t|\pi)$ for the maze at the 10^{th}, 50^{th} and the 100^{th} M-steps. After 50 M-steps a horizon $T \approx 150$ suffices, whereas in earlier M-steps a larger horizon is needed to account for the inferior policy.

2.4 Cut-off time for an infinite horizon

If T is assumed finite and known then our framework can be readily implemented. However, for T infinite one has to select a point at which to terminate the summation in (8). Although our framework doesn't presently yield a formal method, we may use the time marginal $\hat{p}(t|\pi)$ to gain an indication of a suitable cut-off point. To demonstrate the cut-off effect, we consider a simple maze navigation problem where the agent has to learn to traverse a maze from an initial state to a goal state, see figure 2a. The agent has four actions available; up, down, right and left. If the agent moves into a wall it remains in its current state. The environment is stochastic with any action resulting in any of the other actions being performed with probability 0.05. The discount factor is set to $\gamma = 0.95$ and the horizon is set to $T = \infty$. The goal is a sink state from which the agent cannot exit. As the agent remains in the goal state once it has reached it, one would expect the time marginal to be unimodal with the mode being the most likely time for the agent to reach the goal, which is indeed the case, see figure 2b. Therefore a suitable effective horizon is some time after the mode such that the discount factor has significantly reduced the time marginal.

3 Deterministic policies

For every finite MDP the set of optimal policies contains a policy that is deterministic[8]. Indeed, in light of this many classical MDP solution methods restrict their search to deterministic policies, such as policy iteration and value iteration. With this in mind we restrict the policy space to deterministic policies where $\pi_{a,x} = \delta\left(a, a^*(x)\right)$, and run through the same procedure as in section (2.1). The

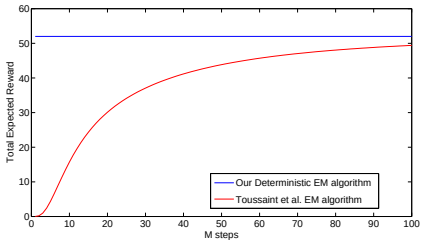


Fig. 3. Our deterministic policy EM algorithm compared with the EM algorithm of [9] on the maze problem in figure 2. The discount factor was set to $\gamma = 1$ and the horizon was set to $T = 100$. The algorithms each performed 100 M-steps and were initialized with the same uniform policy, the total expected utility is as given in (2).

function $a^*(x = \mathsf{x}) = \mathsf{a}$ maps a state x to an single action a and we need to find the mapping that optimises the energy. Expressed in this form the energy becomes

$$E(a^*) = \sum_{t=1}^{T} \sum_{\tau=1}^{t-1} \sum_{x_{\tau+1}, x_\tau} q(x_{\tau+1}, x_\tau, t) \log p(x_{\tau+1}|x_\tau, a^*(x_\tau)) + \sum_{t=1}^{T} \sum_{x_t} q(x_t, t) \log u(x_t, a^*(x_t)) \quad (12)$$

In contrast to the non-deterministic energy, equation (7), we now have an additional term from the utility. Note that this shows the non-commutative nature of taking the deterministic policy limit and optimising – the additional term from the utility cannot be obtained from taking the deterministic limit of (7), *c.f.* [9]. Our procedure then results in an EM algorithm in the deterministic case, as opposed to the 'greedy' policy iteration approach of [9].

In our EM approach, for each state x we now determine the action a that maximizes the energy, equation (12). Since transition probabilities are stationary, this corresponds to finding for each state x that action a that maximises

$$\sum_{\mathsf{x}'} \left\{ \sum_{t=1}^{T} \sum_{\tau=1}^{t-1} q(x_{\tau+1} = \mathsf{x}', x_\tau = \mathsf{x}, t) \right\} \log p(\mathsf{x}'|\mathsf{x}, \mathsf{a}) + \left\{ \sum_{t=1}^{T} q(x_t = \mathsf{x}, t) \right\} \log u(\mathsf{x}, \mathsf{a}) \quad (13)$$

The E-step for the q-distribution is as before, expect that we also require the two-time marginals $q(x_{\tau+1} = \mathsf{x}', x_\tau = \mathsf{x}, t)$.

Experiments We compare the convergence of our deterministic EM algorithm with the non-greedy EM algorithm of [9] on solving the problem in figure 2a with $\gamma = 1$ and horizon $T = 100$. As can be seen in figure 3 our algorithm converges to the optimal policy after the first M-step, where as the stochastic policy EM algorithm of [9] has slower convergence. For comparisons we also ran policy iteration on this problem and noted that it too converged after the first policy update.

3.1 Deterministic transitions

The M-step updates in the EM algorithm characteristically 'freeze', in a deterministic or near-deterministic observation distribution, leading to extremely small increases in the log-likelihood. This problem occurs in our EM approach when the transitions and the policy are both deterministic[2]. In this case all the weight of the q-distribution is put onto the single state-action trajectory that is dictated by the policy and the transition, and the M-step performs the trivial update $\pi^{new} = \pi^{old}$. To counter this problem it is possible to add 'antifreeze' to the environment, rendering it non-deterministic, and then solve the MDP in this new environment. For each state we define the new transition $p_\epsilon(x'|x,a)$ as a convex combination of the transition with a distribution

$$p_\epsilon(x'|x,a) = (1-\epsilon)p(x'|x,a) + \epsilon\Gamma_x(x') \tag{14}$$

where $\epsilon \in [0,1)$ and $\Gamma_x(x')$ is an arbitrary probability distribution and then solve the MDP $\langle \mathcal{X}, \mathcal{A}, U, p_\epsilon \rangle$. The idea behind this is encourage 'exploration' during the E-step and therefore enable the algorithm to escape local minima, similar to ϵ-greedy policies used in various Mote Carlo solution methods to MDPs [8]. For completeness we explain below how the above technique can be both theoretically and practically justified.

4 Antifreeze for EM

4.1 Standard EM learning

To explain the general problem of freezing in EM and a possible resolution, consider a distribution of the form

$$p(v|\theta) = \sum_h p(v|h,\theta)p(h)$$

for which our task is to find the θ that maximises $p(v|\theta)$, given an observed value for v. Treating h as a hidden variable, we may apply the EM algorithm for which the E-step is

$$q(h|\theta_{old}) \propto p\left(v|h,\theta_{old}\right) p(h)$$

and the M-step sets

$$\theta_{new} = \operatorname*{argmax}_\theta \left\langle \log p(v,h|\theta) \right\rangle_{p(h|\theta_{old})} = \operatorname*{argmax}_\theta \left\langle \log p(v|h,\theta) \right\rangle_{p(h|\theta_{old})}$$

since $p(h)$ is independent of θ. For a deterministic observation distribution $p\left(v|h\right) = \delta\left(v, f(h|\theta)\right)$ for some function $f(h|\theta)$ with parameters θ, we have

$$p(h|\theta_{old}) \propto \delta\left(v, f(h|\theta)\right) p(h)$$

[2] For deterministic transitions but a stochastic policy, EM freezing is less problematic.

so that the M-step sets

$$\theta_{new} = \operatorname*{argmax}_{\theta} \langle \log \delta\left(v, f(h|\theta)\right) \rangle_{p(h|\theta_{old})}$$

Since $p(h|\theta_{old})$ is zero everywhere except that h for which $v = f(h|\theta)$, then the energy is negative infinity for $\theta \neq \theta_{old}$ and zero when $\theta = \theta_{old}$. Hence zero is the optimum of the energy, corresponding to a frozen update. This situation occurs in practice, and has been noted in particular in the context of Independent Component Analysis[7] although, as explained here, the phenomenon is quite general. One can attempt to heal this behaviour by deriving an EM algorithm for the distribution

$$p_\epsilon(v|h,\theta) = (1-\epsilon)p\left(v|h,\theta\right) + \epsilon n(h), \qquad 0 < \epsilon < 1$$

where $n(h)$ is an arbitrary 'antifreeze' distribution on the hidden variable h. The original deterministic model corresponds to $p_0(v|h,\theta)$. Hence

$$p_\epsilon(v|\theta) \equiv \sum_h p_\epsilon(v|h,\theta)p(h) = (1-\epsilon)p(v|\theta) + \mathsf{const.}$$

so that applying antifreeze preserves the optima of $p(v|\theta)$ at the same locations as those of $p_\epsilon(v|\theta)$. An EM algorithm for $p_\epsilon(v|\theta)$, $0 < \epsilon < 1$ satisfies

$$p_\epsilon(v|\theta_{new}) - p_\epsilon(v|\theta_{old}) = (1-\epsilon)\left[p_0(v|\theta_{new}) - p_0(v|\theta_{old})\right] > 0$$

which implies $p_0(v|\theta_{new}) - p_0(v|\theta_{old}) > 0$. Hence the EM algorithm for the non-deterministic case $0 < \epsilon < 1$ is guaranteed to increase the likelihood under the deterministic model $p_0(v|\theta)$ at each iteration (unless we are at convergence). Note $n(h)$ can be chosen arbitrarily at each iteration of the EM algorithm, which can help escape local minima.

4.2 Maximising utility

To translate the antifreeze idea into the maximum utility problem, consider an objective

$$F(\theta) = \sum_x u(x)p(x|\theta)$$

for a positive function $u(x)$ with our task being to maximise F with respect to θ. An EM style bounding approach can be derived by defining the auxiliary distribution

$$\tilde{p}(x|\theta) = \frac{u(x)p(x|\theta)}{F(\theta)} \tag{15}$$

so that by considering $\mathrm{KL}(q(x)|\tilde{p}(x))$ for some variational distribution $q(x)$ we obtain the bound

$$\log F(\theta) \geq -\left\langle \log q(x) \right\rangle_{q(x)} + \left\langle \log u(x) \right\rangle_{q(x)} + \left\langle \log p(x|\theta) \right\rangle_{q(x)}$$

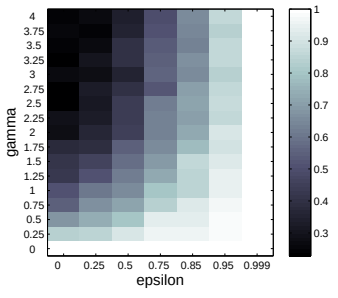


Fig. 4. For each γ, ϵ pair we ran 500 experiments and plot in the figure the fraction of times the correct optimum value is returned by the EM procedure. As γ increases the distribution $p(v|\theta)$ tends to a deterministic distribution and EM optimisation of θ fails. This corresponds to the case $\epsilon = 0$. As we increase ϵ noise more noise is included in the process and the EM algorithm succeeds. Note that for a truly deterministic environment $\gamma = \infty$ a value of $\epsilon < 1$ still suffices, see figure 5.

The M-step states that the optimal q distribution is given by

$$q(x) = \tilde{p}(x|\theta_{old})$$

At the E-step of the algorithm the new parameters θ_{new} are given by maximising the 'energy' term

$$\theta_{new} = \quad \underset{\theta}{\operatorname{argmax}} \left\langle \log p(x|\theta) \right\rangle_{\tilde{p}(x|\theta_{old})}$$

For a deterministic distribution

$$p(x|\theta) = \delta\left(x, f(\theta)\right)$$

the E-step fails since the energy is negative infinity unless $\theta_{new} = \theta_{old}$, in which case the energy is zero. We can attempt to heal this by using the alternative objective

$$F_\epsilon(\theta) = \sum_x u(x) p_\epsilon(x|\theta)$$

with

$$p_\epsilon(x|\theta) = (1-\epsilon) p\left(x|\theta\right) + \epsilon n(x), \qquad 0 \leq \epsilon \leq 1$$

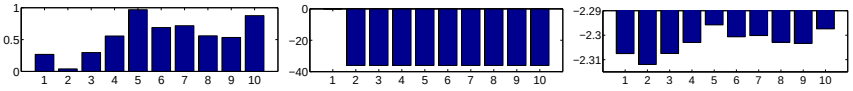
and an arbitrary distribution $n(x)$. Our task is to maximise F with respect to θ. Since

$$F_\epsilon(\theta) = (1-\epsilon) F_0(\theta) + \epsilon \sum_x n(x) u(x)$$

it is clear that $F_\epsilon(\theta)$ has the same optimum as $F_0(\theta)$. Furthermore, since

$$F_\epsilon(\theta_{new}) - F_\epsilon(\theta_{old}) = (1-\epsilon)\left[F_0(\theta_{new}) - F_0(\theta_{old})\right]$$

provided that for $\epsilon > 0$ we can find a θ_{new} such that $F_\epsilon(\theta_{new}) > F_\epsilon(\theta_{old})$, then necessarily $F_0(\theta_{new}) > F_0(\theta_{old})$. Using this result, we may derive an EM-style algorithm that guarantees to increase $F_\epsilon(\theta)$ (unless we are already at the



(a) (b) (c)

Fig. 5. Maximising a utility $L(\theta) = \log \sum_{s=1}^{10} u(s)p(s|\theta)$, where $p(s|\theta)$ is deterministic, placing all its mass in the state $s = \theta$. Here $u(s)$ is given, being positive and drawn at random. The task is to find the optimal θ, which is equivalent in this case to finding the state s that maximises the given $u(s)$. (a) True utility $L(\theta)$ as a function over the 10 θ values. The optimal state is $\theta = 5$. (b) The energy for $\epsilon = 0$ and $\theta^{old} = 1$. The energy is $-\log \infty$ (cut off here at -36) for all but $\theta = \theta^{old} = 1$, where the energy is zero, displaying the characteristic EM freezing. (c) Energy of modified distribution using $\epsilon = 0.99$ and $\theta^{old} = 1$. The new energy has the optimum at the correct place, $\theta = 5$.

optimum) for $\epsilon > 0$ and can therefore guarantee to increase $F_0(\theta)$. To do so we use

$$\tilde{p}_\epsilon(x|\theta) \equiv \frac{u(x)p_\epsilon(x|\theta)}{F_\epsilon(\theta)}$$

in place of equation (15), and then derive an EM algorithm as before.

Applying antifreeze on a toy problem To demonstrate that the effect of adding on noise to the deterministic distribution is non-trivial and can heal the EM algorithm, we carried out a simple experiment, see figure 4. We define a distribution[3] $p(s|\theta) \propto \exp(\gamma \mathbb{I}[s = \theta])$ over the states $s = 1, \ldots, 5$. For each experiment the utility for each state $u(s)$, $s = 1, \ldots, 5$ is drawn from a uniform distribution between 0 and 1. The task is to find θ that maximises $\sum_s u(s)p(s|\theta)$ using an EM style algorithm. To attempt to resolve 'freezing' we added uniform noise by an amount ϵ to the distribution $p(s|\theta)$. In figure 4 we compute the number of times that starting from a random starting state we will, under EM, converge to the correct optimum state. As we can see, for no noise added, $\epsilon = 0$, and in a deterministic limit (γ large) we are in the optimum state only 20% of the time, since no updating occurs, and we start in the correct state with probability 0.2. As we increase ϵ, EM unfreezes and we begin to find the correct optimum. One may have the impression that for a truly deterministic $p(s|\theta)$ we would need to set $\epsilon = 1$ to unfreeze EM and thereby destroy the problem in the process. To show that this is not the case, consider the example in figure 5 which considers a truly deterministic $p(s|\theta)$ yet, by applying antifreeze with $\epsilon < 1$, we find the optimum at the correct place.

[3] This γ is not to be confused with discounting.

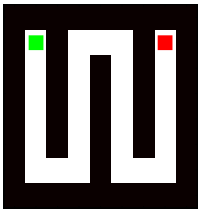


Fig. 6. Maze considered in the deterministic transition and deterministic policy experiments. The colour scheme is the same as previously, that is the walls are black, the initial state is green (top left corner) and the goal state is red (top right corner), with the remaining states in white. The discount factor was set to $\gamma = 1$ and the horizon set to $T = 40$. There are a total of 27 states.

4.3 Antifreeze on an MDP

To illustrate the validity of the 'antifreeze' method in an MDP setting we consider the simple maze problem in figure 6. The transitions are deterministic and the policy is initialised in the worst possible way, taking an action that moves in the opposite direction of the optimal policy, *e.g.* when the agent is in the initial state it will move upwards instead of downwards. Since the environment is deterministic, the normal deterministic EM algorithm would perform trivial updates on this problem and freeze. When implementing antifreeze one has to select the amount of noise and the form of the noise distribution. In the experiments we use an antifreeze distribution $\Gamma_x(x')$ to be uniform for all states that satisfy the condition $\beta_T(x) = 0$, *i.e.* states that have zero probability of receiving a reward under the current policy. The transitions of the remaining states where left unchanged. When adding noise to the transitions ϵ was set to 0.35. The results of the experiment are shown in figure 7 where we can see that our algorithm converged in a single M-step. We also ran the EM algorithm of [9] and policy iteration on this maze problem with the transition probabilities. Policy iteration converges to the optimal policy after roughly 20 policy updates and the EM algorithm of [9] converges more slowly (since it does not explicitly seek a deterministic policy). It should be noted that policy iteration also converges quickly if we add a small amount of noise to the transitions.

5 POMDPs

In a Partially Observed MDP (POMDP) the agent no longer has complete knowledge of its state but instead has only a *belief* of its state[5]. The agent's belief is periodically updated using information gathered through observations $o_{1:T}$. The agent then makes decisions based on its belief and the present observation. Whilst in the MDP the policy is a function of the state, in the POMDP case only a distribution b of the state is known, and optimally the policy needs to be a function of this distribution, rather than the state itself. We do not deal here with the full POMDP case, for which the belief corresponds to the filtered distribution given past observations and actions[5]. Instead we follow the model introduced in [9] which we describe below.

Instead of the true belief we use an auxiliary variable with a fixed small number of states, which we also call the belief and denote with b, that is used to

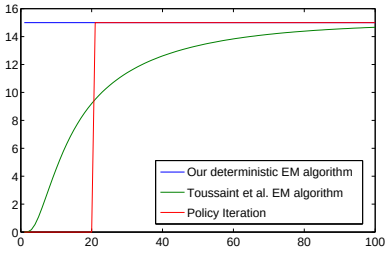


Fig. 7. Antifreeze experiment for a deterministic policy and environment, figure 6. We compare our deterministic EM antifreeze algorithm, $\epsilon = 0.35$ to the EM algorithm of [9] and policy iteration. The two EM algorithms each performed 100 M-steps and were initialized with the same deterministic policy, the total expected utility is as given in (2).

capture the agent's previous experience. We therefore make the policy a function of the current observation and belief, $p(a_t|b_t, o_t)$. As we are no longer dealing with the true belief we now have to introduce a model for the belief transitions, $p(b'|b, o)$. The use of a finite set of belief states can be considered a surrogate for the true filtered belief distribution, as described in [9].

We briefly derive the updates of the belief transitions and then deal with the non-deterministic and deterministic policy updates separately. Following the procedure in section (2) we have the following bound on the expected utility of the POMDP given a policy π,

$$\log u(\pi) \geq -H(q) + \langle \log u(x_t, a_t) \rangle_q + \langle \log p(x_{1:t}, b_{1:t}, o_{1:t}, a_{1:t}, t|\pi) \rangle_q \tag{16}$$

where the probability $p(x_{1:t}, b_{1:t}, o_{1:t}, a_{1:t}, t|\pi)$ is given by

$$p(o_t|x_t)p(x_1)p(b_1) \prod_{\tau=1}^{t-1} p(x_{\tau+1}|x_\tau, a_\tau)p(a_\tau|b_\tau, o_\tau)p(b_{\tau+1}|b_\tau, o_\tau)p(o_\tau|x_\tau)$$

Since we wish to maximise the bound (16) w.r.t $p(b'|b, o)$ we isolate the belief transition terms, which we denote collectively as λ, to obtain the pertinent energy

$$E(\lambda) = \sum_{t=1}^{T} \sum_{\tau=1}^{t-1} \sum_{b_{\tau+1}, b_\tau, o_\tau} q(b_{\tau+1}, b_\tau, o_\tau, t) \log p(b_{\tau+1}, b_\tau, o_\tau) \tag{17}$$

Making the assumption that the belief transitions are stationary then the time dependence of these terms can be removed and the M-step results in the updates

$$p(b'|b, o) \propto \sum_{t=1}^{T} \sum_{\tau=1}^{t-1} q(b_{\tau+1} = b', b_\tau = b, o_\tau = o, t) \tag{18}$$

In the case of the non-deterministic policy updates one runs through the same argument for the policy terms and obtains updates of the form

$$\pi_{b,o} \propto \sum_{t=1}^{T} \sum_{\tau=1}^{t} q(b_\tau = b, o_\tau = o, t) \tag{19}$$

Whilst in the case of the full POMDP there is no reason to expect the optimal policy to be deterministic, our POMDP model is similar to a MDP in that decisions are made on the basis of a discrete variable, rather than a distribution. In this case one may expect the optimal policy to be deterministic. With this in mind we again restrict the policies to be deterministic and re-derive the M-step, giving the following updates

$$a(b,o) = \operatorname*{argmax}_a \sum_x q(x,b,o) \log u(x,a(b,o)) + \sum_{x',x} q(x',x,b,o) \log p(x'|x,a(b,o))$$

where

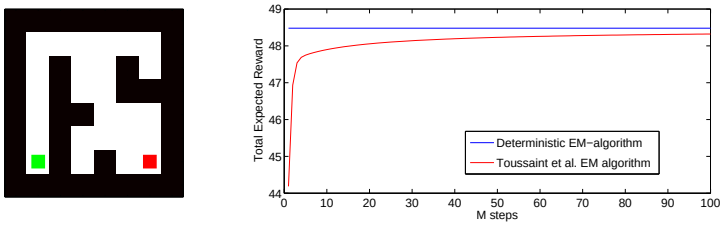
$$q(x,b,o) = \sum_{t=1}^{T} q(x_t = x, b_t = b, o_t = o, t)$$

$$q(x',x,b,o) = \sum_{t=1}^{T} \sum_{\tau=1}^{t-1} q(x_{\tau+1} = x', x_\tau = x, b_\tau = b, o_\tau = o, t)$$

5.1 Experiments

We compare our deterministic policy POMDP EM algorithm with the non-deterministic policy algorithm of [9] (for which no deterministic algorithm was previously known). The maze problem is depicted in figure 8. Unlike the previous maze problem the agent is unaware of its position in the maze and can only make decisions on the current observation (the configuration of walls that are currently adjacent to the agent) and its belief in its position. The observations are with respect to a constant 'northward' orientation, *i.e.* the agent is always assumed to be facing northward, so the observation 0111 at the initial state means no wall north, wall east, wall south, wall west. The actions that the agent can make are moving North, South, East or West, where a movement into a wall will result in the agent remaining in the same state. The environment is stochastic in the sense that any action will result in another action being taken with probability 0.05.

The results of the experiment are shown in figure 8, where we can see that our EM algorithm converges in the first M-step, supporting the intuition that the optimal policy in this framework is indeed deterministic. We also ran the EM algorithm of [9] on this problem where we can see that it has still to converge after 100 M-steps, although it does display rapid progress in the initial stages. Whilst we have demonstrably some success with our deterministic POMDP approach it should be noted that our experience is that our algorithm can be more fragile that the stochastic variant [9]. A pragmatic approach in practice therefore is to alternate between the deterministic and stochastic policy algorithms once a local optimum has been reached.



(a) (b)

Fig. 8. (a) Maze for the POMDP experiments for deterministic EM algorithms. The initial state is in the bottom left corner (green), with the goal state in the bottom right corner (red). The rest of the maze consists of walls (black) and remaining possible positions (white). There are in total 26 states with 11 different distinct observations (generally one could expect there to be up to 16 different possible observations in an arbitrary maze). (b) Comparison of convergence of our deterministic policy POMDP algorithms with that of [9]. The discount factor was set $\gamma = 1$, and the horizon limit was set $T = 50$.

6 Discussion

We discussed a framework for policy learning in MDPs that treats the problem as learning in a related probabilistic model. Our approach draws heavily on previous work [9] though the framework is simpler in that no auxiliary reward variables are required and the method directly handles the case of no discounting. In particular we derived a true EM style procedure for MDPs and POMDPs restricted to deterministic policies, which differ from the standard policy iteration updates previously derived [9] for the MDP, and introduces a novel deterministic policy update in the case of the POMDP. Treating the deterministic policy case correctly is important since in the MDP the optimal policy is deterministic. In future extensions of this work our aim is to attempt to solve large scale MDPs using techniques in approximate inference, for which a correct variational treatment of the deterministic case is a required starting point.

An important limitation of all EM approaches is that in a deterministic environment (in a standard EM problem this corresponds to the observation distribution being deterministic, and in the MDP case the analog is that the environment transitions are deterministic) EM freezes, and no updating occurs. This also happens for low noise environments. We introduced a principled 'antifreeze' method that potentially heals this problem by considering a modified environment being a convex combination of the true environment and a noise distribution.

Software Demonstration software for solving (PO)MDPs for both deterministic and stochastic policies may be found at `www.cs.ucl.ac.uk/staff/D.Barber`.

Acknowledgement We would like to thank Marc Toussaint for helping clarify the relationship between his and our framework.

References

1. P. Dayan and G. E. Hinton. Using Expectation-Maximization for Reinforcement Learning. *Neural Computation*, 9:271–278, 1997.
2. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc*, 39:1–38, 1977.
3. T. Hoffman, N. de Freitas, A. Doucet, and J. Peters. An Expectation Maximization Algorithm for continuous Markov Decision Processes with Arbitrary Rewards. *AISTATS*, 2009.
4. F. V. Jensen and T. D. Nielson. *Bayesian Networks and Decision Graphs*. Springer Verlag, second edition, 2007.
5. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domain. *Artificial Intelligence*, 101:99–134, 1998.
6. H. J. Kappen. An introduction to stochastic control theory, path integrals and reinforcement learning. In *Proceedings 9th Granada seminar on Computational Physics: Computational and Mathematical Modeling of Cooperative Behavior in Neural Systems*, volume 887, pages 149–181. American Institute of Physics, 2007.
7. K. B. Petersen and O. Winther. The EM algorithm in independent component analysis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 5, pages 169–172, 2005.
8. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
9. M. Toussaint, S. Harmeling, and A. Storkey. Probabilistic inference for solving (PO)MDPs. Research Report EDI-INF-RR-0934, University of Edinburgh, School of Informatics, 2006.
10. M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.