An Associative State-Space Metric for Learning in Factored MDPs

Pedro Sequeira, Francisco S. Melo, and Ana Paiva

INESC-ID and Instituto Superior Técnico, Technical University of Lisbon
Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Portugal
pedro.sequeira@gaips.inesc-id.pt, {fmelo,ana.paiva}@inesc-id.pt

Abstract. In this paper we propose a novel *associative metric* based on the classical conditioning paradigm that, much like what happens in nature, identifies associations between stimuli perceived by a learning agent while interacting with the environment. We use an associative tree structure to identify associations between the perceived stimuli and use this structure to measure the degree of similarity between states in factored Markov decision problems. Our approach provides a *state-space metric* that requires no prior knowledge on the structure of the underlying decision problem and is designed to be learned online, *i.e.*, as the agent interacts with its environment. Our metric is thus amenable to application in reinforcement learning (RL) settings, allowing the learning agent to generalize its experience to unvisited states and improving the overall learning performance. We illustrate the application of our method in several problems of varying complexity and show that our metric leads to a performance comparable to that obtained with other well-studied metrics that require full knowledge of the decision problem.

1 Introduction

Associative learning is a paradigm from the field of behaviorism that posits that learning occurs whenever a change in behavior is observed [1]. Classical conditioning is one of the best-known associative learning paradigms. It is one of the most basic survival tools found in nature that allows organisms to expand the range of contexts where some of their already-known behaviors can be applied. By associating co-occurring stimuli from the environment, the organism can activate innate phylogenetic responses (*e.g.*, fight or flight responses) to new and previously unknown situations.

In this paper, we leverage this idea to reinforcement learning (RL). RL agents explore their environment and gather information that allows them to learn the best actions to take in different situations. Many classical RL methods, such as Q-learning, allow the agent to successively estimate how good each action is in every state, eventually conveying to the agent the information necessary to select only the best actions in all states. This typically requires the agent to experience every action in every state a sufficient number of times [2]. This need for "sufficient" visits to every state-action pair is often impractical, particularly in large environments, and several general approaches have been proposed to

L. Correia, L.P. Reis, and J. Cascalho (Eds.): EPIA 2013, LNAI 8154, pp. 163–174, 2013.

mitigate this need, relying mostly on some form of function approximation (we refer to [3] for references and discussion).

However, certain scenarios present some particular structure that can be leveraged by the learning algorithm to improve the learning performance—namely, by alleviating the requirement of sufficient visits to every state-action pair. For example, in scenarios where the state is described by a finite set of state-variables (*i.e.*, where the state is *factored*), it is often possible to use this structure to improve the efficiency of RL [4]. This is particularly true if many of the state-variables are irrelevant for the task that to be learned, and it is possible to improve the learning performance by identifying such irrelevant state-variables, allowing the learning agent to focus only on those that are relevant [5,6].

Our approach builds on all aforementioned ideas. We introduce a method that allows the learning agent to identify *associations* between perceived stimuli during its interaction with the environment. Specifically, given a learning scenario with a factored state space, we use a pattern mining technique to build an *association tree* that identifies the occurrence of frequent *patterns* of state-variables (henceforth referred as *stimuli*) [7]. These associations are similar in spirit to those that natural organisms identify in their interaction with the environment, and are then used by the agent to build a metric that identifies two states as being "close" if they share multiple/frequent stimuli. This metric is learned online and combined with Q-learning, as proposed in [8] to improve the learning performance of our agents and use current information to update the value of states that are considered *similar* according to the associative metric.

The main contribution of our approach is to provide a general-purpose state-space metric that requires *no prior knowledge* on the structure of the underlying decision problem. The associative tree and the similarity metric are both learned online, *i.e.*, while the agent is interacting with its environment, making it particularly amenable to use in a reinforcement learning setting. We illustrate the application of our method in several factored Markov decision processes (MDPs) of varying complexity and show that our metric leads to a performance comparable to that obtained when using well-studied metrics from the literature [9].

2 Background

In this section we introduce the necessary background on both the biological and computational concepts that will be used throughout the paper.

2.1 Reinforcement Learning

The field of *reinforcement learning* (RL) addresses the general problem of an agent faced with a sequential decision problem [2]. By a process of trial-and-error, the agent must learn a "good" mapping that assigns states to actions. Such mapping determines how the agent acts in each possible situation and is commonly known as a *policy*. In a sense, reinforcement learning is the computational counterpart to the notion of *reinforcement* used in operant conditioning and behavior analysis [2,10].

RL agents can be modeled using *Markov decision processes* (MDPs). At every step t, the agent/environment is in state $X(t) = x$, with $x \in \mathcal{X}$ and chooses an action $A(t) = a$, with $a \in \mathcal{A}$. Both $\mathcal{X}$ and $\mathcal{A}$ are assumed finite. Given that $X(t) = x$ and $A(t) = a$, the agent/environment transitions to state $y \in \mathcal{X}$ with probability given by

$$\mathsf{P}(y \mid x, a) \triangleq \mathbb{P}\left[X(t+1) = y \mid X(t) = x, A(t) = a\right]$$

and receives a reward $r(x, a)$, and the process repeats. The agent must choose its actions so as to gather as much reward as possible, discounted by a positive discount factor $\gamma < 1$. Formally, this corresponds to maximizing the value

$$v = \mathbb{E}\left[\sum_t \gamma^t r(X(t), A(t))\right], \tag{1}$$

where, as before, $X(t)$ and $A(t)$ denote the state and action at time-step t, respectively. The reward function r implicitly encodes the *task* that the agent must accomplish. It is a well-known fact that in (finite) MDPs it is possible to find a *policy* $\pi^* : \mathcal{X} \to \mathcal{A}$ maximizing the value in (1). Associated with the optimal policy π^* is the *optimal Q-function*,

$$Q^*(x, a) = \mathbb{E}\left[\sum_t \gamma^t r(X(t), A(t)) \mid X(0) = x, A(0) = a\right],$$

from which the optimal policy can easily be computed [2].

In many MDPs the state $X(t)$ can be described using a finite set of *state features* $X_i(t), i = 1, \ldots, n$, each taking values in some feature space $\mathcal{X}_i$. The state-space thus corresponds to the cartesian product $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$. The structure exhibited by such *factored MDPs*, both in terms of transition probabilities and reward function, can often be exploited, leading to more efficient solution methods [4,11]. The computational gains can be particularly noteworthy if many of the state-features are *irrelevant* for the underlying task to be solved by the agent. In fact, it is possible to greatly improve the performance of solution methods by identifying such irrelevant state-features and focusing only on those that are relevant [5,6]. In this paper, we refer to an element $x = (x_1, \ldots, x_n)$ as a state and to an element $x_i \in \mathcal{X}_i$ as a *stimulus*. We consider stimuli as *categorical nominal data*, *i.e.*, variables that describe discrete values.

If the MDP model is known, the function Q^* can easily be computed using, for example, dynamic programming. However, in RL settings, the dynamics P and reward r of the MDP model are typically unknown. The agent must thus *learn* Q^* through interactions with its environment. This can be achieved using, for example, the *Q-learning algorithm* [10], that updates the estimate for Q^* as

$$\hat{Q}(x(t), a(t)) \leftarrow (1 - \alpha_t)\hat{Q}(x(t), a(t)) + \alpha_t\big(r(t) + \gamma \max_b \hat{Q}(x(t+1), b)\big), \tag{2}$$

where $x(t)$ and $a(t)$ are the state and actions experienced (sampled) at time t, $r(t)$ is the received reward and $x(t+1)$ is the subsequent state. Q-learning is

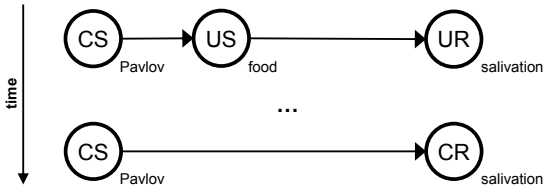


Fig. 1. Example of classical conditioning in a dog, inspired in Pavlov's experiments [12]

guaranteed to converge with probability 1 as long as every state-action pair is visited infinitely often and the step-size sequence, $\{\alpha_t\}$, verifies standard stochastic approximation conditions.

The need for infinite visits to every state-action pair is unpractical in many situations, and several general approaches have been proposed to mitigate this need. In this paper we adopt a simple technique proposed in [8], where Q-learning is combined with a *spreading function* that "spreads" the estimates of the Q-function in a given state to neighboring states. Formally, given a similarity function $\sigma_t(x, y)$ that measures how close two states x and y are, the Q-learning with spreading update is given by

$$\hat{Q}(x, a(t)) \leftarrow (1-\alpha_t)\hat{Q}(x, a(t)) + \alpha_t \sigma_t(x, x(t))\big(r(t) + \gamma \max_b \hat{Q}(x(t+1), b)\big). \quad (3)$$

As discussed in [8], convergence of Q-learning with spreading to the optimal Q-function can be guaranteed as long as the spreading function σ_t converges to the Kronecker delta-function at a suitable rate.[1]

2.2 Classical Conditioning

Figure 1 illustrates a typical setting for a classical conditioning experimental procedure. In a first phase, known as *initial pairing* or *training*, an organism's biologically significant *unconditioned stimulus* (US) is paired with a neutral, biologically meaningless stimulus, called the *conditioned stimulus* (CS) [1,12]. The US—for example food or an electrical shock,—reflexively evokes innate, automatic unconditioned responses (UR)—for example, salivating or freezing. The neutral CS can be any event that does not result in an overt behavioral response from the organism under investigation (*e.g.*, the sound of a bell, a light or even the presence of a person). In a second phase (*testing*), and after a few pairings between the US and CS, have occurred, the experimenter measures the level of response from the organism when exposed to the CS alone, with no US being presented. The experimenter typically observes a change in response from the organism in the presence of the CS, which now evokes a conditioned response (CR) similar to the UR evoked by the US.

Following the example in Fig. 1, the presence of Pavlov alone made the dogs start salivating in anticipation of food delivery. This change in response is due to the development of an *association* between a representation of the CS and one of the US, arising from the *co-occurrence of both stimuli*. This is the main idea

[1] Actually, the algorithm described in [8] also considers spreading across actions. In this paper we address only spreading across states.

behind Pavlov's *stimulus substitution theory* [12], where the CS "substitutes" the US in evoking the reflexive response.

The evolutionary advantages behind such associative mechanism could be the ability of organisms to broaden the contexts where they apply some advantageous response, and to anticipate the biological significance of co-occurring events [13]. By determining associations between stimuli in the environment, animals are able to: (i) recognize contexts (states) of the environment and thus anticipate rewards or punishments and consequences of behavior that are similar to those observed in previous interactions; (ii) integrate information from previous observations with new, never before experienced stimuli.

Inspired in these ideas from classical conditioning, our learning approach: (i) spreads action and reward information (the Q-values) between similar states; (ii) integrates information in new, unknown states, from the Q-values of previously experienced similar states.

3 Associative Metric for RL

In this section we introduce a new associative metric to be used in factored MDPs. We take inspiration in the classical conditioning paradigm introduced in the previous section, and port some of the underlying principles into an RL context, effectively improving the performance of RL agents.

To better explain our learning procedure let us consider a behavior phenomenon associated with the classical conditioning paradigm known as *secondary conditioning* or *sensory preconditioning* as an example to follow throughout this section. Secondary conditioning takes place whenever a CS (CS1) that is trained to predict some US is paired with a different CS (CS2), either before or after CS1 and US are paired. By means of this secondary association, CS2 also becomes associated with the US value through its association with CS1, and ends up evoking the same kind of CR [14]. Figure 2 illustrates an example of the secondary conditioning phenomenon, where, for explanatory purposes, we consider that the stimuli come from different perceptual modalities.

Biologically speaking, after being trained with sound-shock pairings followed by sound-light pairings, the agent should be able to predict the presence of the shock whenever it perceives the light, even if the two stimuli never co-ocurred. From a more computational perspective, the learning procedure should discover that environmental states involving light and shock are somehow *associated*. In this manner, whatever value is associated with the stimulus "shock" should, to some extent, also be associated with "light", and the outcome of executing similar actions in associated states should, to some extent, be similar.

We can therefore decompose the learning problem into two sub-problems: *identifying associated states* and *using the information* about some experienced states in other (associated) states. Sections 3.1 and 3.2 describe our approach in addressing the first sub-problem, where we propose the combination of a sensory pattern tree and a new associative metric to measure the distance between associated states. In Section 3.2 we discuss how this metric can then be combined with Q-learning with spreading to improve the performance of an RL agent.

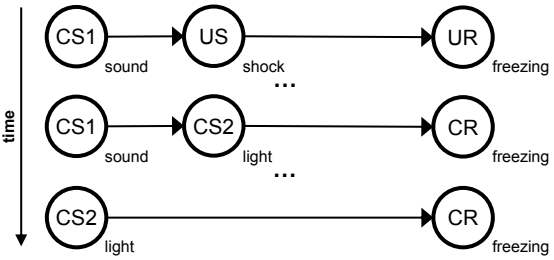


Fig. 2. Example of a *secondary conditioning* phenomenon.

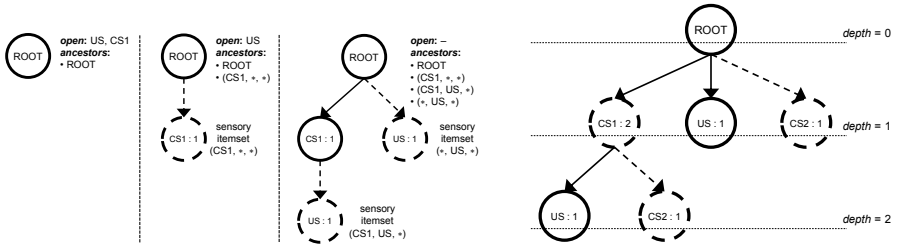


Fig. 3. The construction of an associative sensory tree. The updated and inserted nodes at each step are marked with a dashed line. (a) The steps involving the construction of the tree after an initial observation of state $x(1) = (\mathsf{CS1}, \mathsf{US}, \emptyset)$, where the sensory itemsets associated with each new node are explicitly indicated; (b) Updated tree after observing state $x(2) = (\mathsf{CS1}, \emptyset, \mathsf{CS2})$. The depth of each node is explicitly indicated.

3.1 Sensory Pattern Mining

As we have seen in Section 2.2, one of the fundamental aspects in the classical conditioning paradigm is the ability of individuals to establish associations between the stimuli they perceive. Stimuli that are frequently perceived together are more likely to lead to similar *value* and *outcome* than stimuli that seldom co-occur. Inspired by this idea, our approach aims at endowing learning agents with a mechanism allowing them to determine how "similar" two states are based on how many associated stimuli they share. To determine such associations we follow a method introduced in [7], where an online sensory pattern mining technique is proposed to identify associations between stimuli occurring in the agent's perceptions, while the agent interacts with its environment. This method identifies such associations by incrementally constructing an *associative sensory tree*, using a variation of the FP-growth algorithm [15] commonly used for transactional pattern mining.

We denote a collection of state-elements $\mathbf{s} = \langle x_{i_1}, \ldots, x_{i_k} \rangle$, with $k \leq n$, as a *sensory itemset*. We assume without loss of generality that each feature-space $\mathcal{X}_i, i = 1 \ldots, n$, is an ordered set[2]. The general sensory pattern mining algorithm in [7] dynamically builds a sensory tree as follows:

[2] We note that the specific order of the elements $\mathcal{X}_i \in \mathcal{X}$ is not important, as long as it remains fixed throughout learning. This is a requirement of the tree construction algorithm that guarantees a minimal representation [7].

- At every time-step t, the agent observes state $X(t) = (x_1(t), \ldots, x_n(t))$;
- Given $X(t)$, the algorithm updates the tree by keeping two lists: an "open" list, initially containing all elements $x_i(t) \in X(t)$ to be inserted into the tree; an "ancestor" list containing the nodes in the tree updated so far, which at the beginning of each update contains only the ROOT node (see Fig. 3(a));
- The algorithm then picks one element $x_i(t)$ from the "open" list at a time, ignoring absent elements ($\emptyset$). For each element, a child node $\mathbf{s}$ is created for each node in the "ancestor" list, with counter $n(\mathbf{s}) = 1$. If the child node already exists, its counter is incremented by 1 (see Fig 3(b)). Each new *node* in the tree represents a *sensory itemset*, *i.e.*, a sub-combination of elements obtained from $X(t)$. The nodes' *counter* represents the number of times the corresponding itemset was observed by the agent so far.

Referring back to the example in the beginning of this section, let us consider that we have the state-space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, where $\mathcal{X}_1 = \{\mathsf{CS1}, \emptyset\}$, $\mathcal{X}_2 = \{\mathsf{US}, \emptyset\}$ and $\mathcal{X}_3 = \{\mathsf{CS2}, \emptyset\}$, where $\emptyset$ represents the absence of a particular stimulus. In other words, each state $x \in \mathcal{X}$ is described by the presence or absence of each of the three stimuli $X_i, i = 1, \ldots, 3$. Figure 3 shows the steps involving the construction of the tree when the agent perceives the state $X(1) = x = (\mathsf{CS1}, \mathsf{US}, \emptyset)$ (sound-shock pairing) from the environment followed by state $X(2) = y = (\mathsf{CS1}, \emptyset, \mathsf{CS2})$ (sound-light pairing).

Given the associative sensory tree, one can measure at each time-step the *degree of association* between stimuli in some sensory itemset $\mathbf{s}$, using the *Jaccard index* [16] which can be used to measure the similarity of sample sets. Given the itemset $\mathbf{s} = \langle x_{i_1}, \ldots, x_{i_k} \rangle$, let $\mathrm{d}(\mathbf{s})$ and $n(\mathbf{s})$ denote, respectively, the *depth* of and the *counter* associated with the corresponding node in the tree. For nodes not directly below the ROOT ($d > 1$), the Jaccard index of $\mathbf{s}$ is given by

$$J(\mathbf{s}) = \frac{n(\mathbf{s})}{\sum_{\mathbf{s}_d} (-1)^{\mathrm{d}(\mathbf{s}_d)+1} n(\mathbf{s}_d)}, \tag{4}$$

where the summation is taken over all nodes $\mathbf{s}_d$ in the *dependency tree* of $\mathbf{s}$, *i.e.*, the subtree containing all nodes in the "ancestor" list obtained after introducing itemset $\mathbf{s}$ in the tree.

Returning to our example, we can now calculate the Jaccard index of state x by solving (4):

$$J(\mathbf{s}) = \frac{n(\mathsf{CS1}, \mathsf{US}, *)}{n(\mathsf{CS1}, *, *) + n(*, \mathsf{US}, *) - n(\mathsf{CS1}, \mathsf{US}, *)} = \frac{1}{2}$$

As expected, the index is inferior to 1, as stimulus CS1 also appears in y, where the US is absent.

We conclude by noting that associative sensory trees are variations of *FP-trees*, which are known to provide a compact representation of large transactional databases [15]. Associative sensory trees have an important advantage over FP-trees, since all information necessary to compute the degree of association between stimuli is trivially accessible from the tree (unlike in an FP-tree). We refer to [7] for further discussion.

3.2 Associative Metric for Factored MDPs

To define a metric using the associative sensory tree described in the previous section, we introduce some additional notation that facilitates the presentation. For any state $x \in \mathcal{X}$, let $\mathcal{S}(x)$ denote the set of all sensory itemsets associated with x. This corresponds to the set of all sub-combinations of stimuli in the dependency tree of the sensory itemset $\mathbf{s}$ associated with x.

We are now in position to introduce our state-space metric. Given the sensory tree at time-step t, we consider the distance between two states x and y as

$$d_A(x,y) = 1 - \frac{\sum_{\mathbf{s}\in\mathcal{S}(x)\cap\mathcal{S}(y)} J_t(\mathbf{s})}{\sum_{\mathbf{s}\in\mathcal{S}(x)\cup\mathcal{S}(y)} J_t(\mathbf{s})}. \tag{5}$$

The distance d_A is indeed a proper *metric*, as it can be reduced to the Tanimoto distance [17] between two vectors associated with x and y, each containing the Jaccard indices for the sensory patterns associated with x and y, respectively. Intuitively, the metric in (5) translates the rationale that two states x and y are "similar" if either they share many stimuli and/or many associated stimuli (stimuli that often co-exist).

Having defined the associative metric we can tackle the first problem defined in the beginning of the section and determine whether two states are similar or not. We can define $\mathcal{S}(x) = \{(\mathsf{CS1}, \mathsf{US}, *), (\mathsf{CS1}, *, *), (*, \mathsf{US}, *)\}$ and $\mathcal{S}(y) = \{(\mathsf{CS1}, *, \mathsf{CS2}), (\mathsf{CS1}, *, *), (*, *, \mathsf{CS2})\}$. The distance between x and y in the example can then be calculated from (5) as

$$d_A(x,y) = 1 - \frac{1}{0.5 + 0.5 + 1 + 0.5 + 0.5} = \frac{2}{3}$$

This means that the degree of similarity between the two states is $1/3$. It follows that our proposed model supports the secondary conditioning phenomenon: the light and foot shock stimuli have some degree of association by means of the sound stimulus, although $\mathsf{CS2}$ and US were never observed together by the agent.

Now that we are able to identify similar states we describe how the metric in (5) can be combined online with Q-learning with spreading. In the experiments reported in this paper, we use the spreading function $\sigma_t(x,y) = e^{-\eta_t d_A(x,y)^2}$. The sequence $\{\eta_t\}$ is a slowly increasing value that ensures that σ_t approaches the Kronecker delta function at a suitable rate, and d_A is the metric defined in (5). As seen in Section 2.1, at each time step t the spreading function σ_t uses information from the current state $X(t)$ to update all other states $y \in \mathcal{X}$, depending on the similarity between $X(t)$ and y calculated according to the structure of the sensory tree at t.

3.3 MDP Metrics and Function Approximation in RL

The notion of "similarity" between states has recently been explored in the MDP literature as a means to render solution methods for MDPs more efficient [9,18]. In fact, by identifying "similar" states in an MDP $\mathcal{M}$, it may be possible to construct a smaller MDP $\mathcal{M}'$ that can more easily be solved.

As established in [19], "similarity" between MDP states is best captured by the notion of *bisimulation*. Bisimulation is an equivalence relation $\sim$ on $\mathcal{X}$ in which two states x and y are similar if $r(x,a) = r(y,a)$ for all $a \in \mathcal{A}$ and

$$\mathbb{P}\left[X(t+1) \in U \mid X(t) = x, A(t) = a\right] = \mathbb{P}\left[X(t+1) \in U \mid X(t) = y, A(t) = a\right],$$

where U is some set in the partition induced by $\sim$. Lax bisimulation is a generalization of bisimulation that also accounts for action relabeling. Both bisimulation and lax bisimulation led to the development of several *MDP metrics* in which, if the distance between two states x, y is zero, then $x \sim y$ [9].

While MDP metrics such as the one above were designed to improve efficiency in MDP solution methods, the best grounded MDP metrics—namely, those relying in the so-called Kantorovich metric—are computationally very demanding [9]. Additionally, they require complete knowledge of the MDP parameters, which renders them unsuitable for RL.

Nevertheless, many RL methods using function approximation implicitly or explicitly exploit some state-space metric [20,21]. Metrics with well-established theoretical properties (*e.g.*, the bisimulation metric discussed above) could potentially bring significant improvements to RL with function approximation.

The metric proposed in this paper, being computed online, is suitable for RL. Besides, as our results show, in MDPs with a factored structure, our metric is able to attain a generalization performance that matches that obtained with more powerful metrics (such as the bisimulation metric).

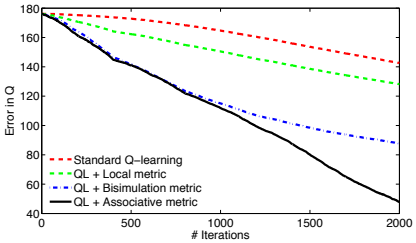
4 Experimental Results

In this section we describe several simple experiments aiming at illustrating the applicability of our method. The experiments show the potential of combining the proposed associative metric with spreading in Q-learning, providing a boost in the agent's performance in several factored MDP problems. The main conclusions stemming from the experiments are analyzed and discussed.

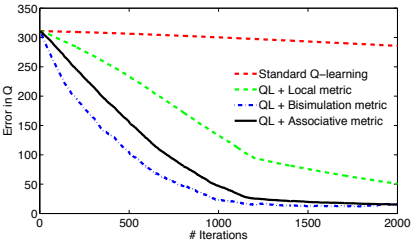
To assess the applicability of our method, we applied Q-learning with spreading using σ_t defined earlier and our associative metric in several factored environments, with a state-space that could be factored into between 1 and 4 factors, with a number of states between 20 and 481, and 5 actions. The transition probabilities between states and the reward function were generated randomly. We present the results obtained in 4 of those environments having, respectively, 20 states (5×4), 60 states ($5 \times 4 \times 3$), 120 states ($5 \times 4 \times 3 \times 2$) and 481 states ($9 \times 7 \times 7$, where the dimension and number of factors was chosen randomly). In all scenarios we use $\gamma = 0.95$ and uniform exploration.

We compare the performance of standard Q-learning with that of Q-learning with spreading using several metrics. In particular, we compare 3 metrics:

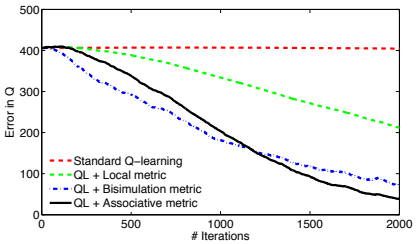
- A *local metric*, d_ℓ, computed from the transition probabilities of the MDP. Given two states $x, y \in \mathcal{X}$, $d_\ell(x,y)$ corresponds to the average number of steps necessary to transition between the two states, which in grid-world scenarios roughly corresponds to the Manhattan distance. The distance between states that do not communicate was set to an arbitrary large constant.



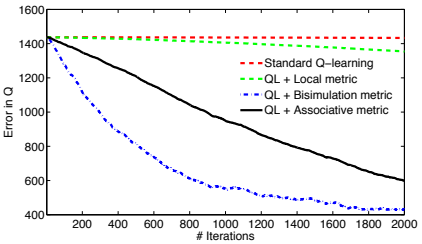
(a) 20 states (2 factors).



(b) 60 states (3 factors).



(c) 120 states (4 factors).



(d) 481 states (3 factors).

Fig. 4. Performance of Q-learning with spreading in different factored scenarios measuring the error in the Q-values. We compare different metrics with varying knowledge of the MDP parameters. Results are averages over 10 independent Monte-Carlo trials.

- A simplified *bisimulation metric*, d_b [9]. The distance d_b is a simplified version of the bisimulation metric that relies on the total variation norm discussed in Section 3.3 and originally proposed in [9, Section 4.2].[3] We note that this is a theoretically sound metric that, however, requires complete knowledge of both P and r.
- The associative metric d_A described in Section 3.2.

For each of the test scenarios, we ran 10 independent Monte-Carlo trials, and evaluated the learning performance of the different methods by comparing the speed of convergence to the optimal Q-function. The parameter η_t was optimized empirically for each metric in each environment so as to optimize the performance of the corresponding method. Figure 4 depicts the average results.

We note, first of all, that our method always outperforms both standard Q-learning and the local metric. The fact that our method learns faster than standard Q-learning indicates that, in these scenarios, the associations between stimuli provide a meaningful way to generalize the Q-values across states. It is also not surprising that our method generally outperforms the local metric, since it implicit assumes that there is some "spacial" regularity that can be used to generalize Q-values across neighboring states. However, this is generally not the case, meaning that in some scenarios the local metric does not provide a significant improvement in performance—see, for example, Figs. 4(a) and (d).

The bisimulation metric, although a simplified from [9], is a metric that takes into consideration both the transition structure and the reward function of the

[3] To simplify, we treat each state as an *equivalence class*. We refer to [9].

MDP. As such, it is not surprising that it allows for good generalization. The fact that our metric performs close to the bisimulation metric in several scenarios—see, for example, Figs. 4(a), (b) and (c)—is, on the other hand, a significant result, since our metric is *learned online*, while the agent interacts with the environment and so uses no prior knowledge on the MDP.

Finally, we note that our metric relies on the factorization of the state-space to build the sensory tree, since the latter is built by associating state-variables that co-occur frequently. In a non-factored MDP, our method would essentially reduce to standard Q-learning. The reliance of our metric on the factorization of the state-space justifies, to some extent, the result in Fig. 4(d). In fact, this corresponds to a large MDP where the "factors" of the state-space are also large. Therefore, not only is the problem larger and, thus, harder to learn, but also our method is able to generalize less than in other more factored scenarios.

5 Concluding Remarks

In this paper we proposed a new state-space associative metric for factored MDPs that draws inspiration from classical conditioning in nature. Our metric relies on identified associations between state-variables perceived by the learning agent during its interaction with the environment. These associations are learned using a sensory pattern-mining algorithm and determine the similarity between states, thus providing a state-space metric that requires no prior knowledge on the structure of the underlying decision problem. The sensory pattern-mining algorithm relies on the *associative sensory tree*, that captures the frequency of co-occurrence of stimuli in the agent's environment.

It is worth mentioning that the size of the associative sensory tree exhibits a worst-case exponential dependence in the number of *state-factors* (not states). However, aside from the memory requirements associated therewith, the structure of the tree is such that the computation of the distance is linear in the number of factors, which is extremely convenient for the online processing of distances. Moreover, as discussed in Section 3.1, the adopted tree representation can safely be replaced by other equivalent representations, such as the FP-tree [15] that, while more efficient in terms of memory requirements, may render the computation of the distance computationally more expensive.

Additionally, we note that the maximal size of the tree is only achieved when *all* the state space has been explored. However, it is in the early stages of the learning process—when little of the state space has been explored—that the use of associative metric may be more beneficial. Our results indicate that the combination of our metric with standard Q-learning does lead to an improved learning performance that is comparable to that obtained with other more powerful metrics that use information both from the transitions and rewards of the MDP. The specific strategy used to integrate the metric with Q-learning (*i.e.*, the decaying spreading function) enforces that when the size of the tree approaches its maximum size, the contribution of the associative metric to learning is generally small. Therefore, limiting the tree size to some pre-specified maximum or using tree-pruning techniques as those discussed in [7] should have little impact on the performance of our proposed method.

Acknowledgments. This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) under project PEst-OE/EEI/LA0021/2013 and the EU project SEMIRA through the grant ERA-Compl /0002/2009. The first author acknowledges grant SFRH/BD/38681/2007 from FCT.

References

1. Anderson, J.: Learning and Memory: An Integrated Approach. Wiley (2000)
2. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
3. Szepesvári, C.: Algorithms for Reinforcement Learning. Morgan & Claypool (2010)
4. Kearns, M., Koller, D.: Efficient reinforcement learning in factored MDPs. In: Proc. 1999 Int. Joint Conf. Artificial Intelligence, pp. 740–747 (1999)
5. Jong, N., Stone, P.: State abstraction discovery from irrelevant state variables. In: Proc. 19th Int. Joint Conf. Artificial Intelligence, pp. 752–757 (2005)
6. Kroon, M., Whiteson, S.: Automatic feature selection for model-based reinforcement learning in factored MDPs. In: Proc. 2009 Int. Conf. Machine Learning and Applications, pp. 324–330 (2009)
7. Sequeira, P., Antunes, C.: Real-time sensory pattern mining for autonomous agents. In: Cao, L., Bazzan, A.L.C., Gorodetsky, V., Mitkas, P.A., Weiss, G., Yu, P.S. (eds.) ADMI 2010. LNCS, vol. 5980, pp. 71–83. Springer, Heidelberg (2010)
8. Ribeiro, C., Szepesvári, C.: Q-learning combined with spreading: Convergence and results. In: Proc. Int. Conf. Intelligent and Cognitive Systems, pp. 32–36 (1996)
9. Ferns, N., Panangaden, P., Precup, D.: Metrics for finite Markov decision processes. In: Proc. 20th Conf. Uncertainty in Artificial Intelligence, pp. 162–169 (2004)
10. Watkins, C.: Learning from delayed rewards. PhD thesis, King's College, Cambridge Univ. (1989)
11. Guestrin, C., Koller, D., Parr, R., Venkataraman, S.: Efficient solution algorithms for factored MDPs. J. Artificial Intelligence Research 19, 399–468 (2003)
12. Pavlov, I.: Conditioned reflexes: An investigation of the physiological activity of the cerebral cortex. Oxford Univ. Press (1927)
13. Cardinal, R., Parkinson, J., Hall, J., Everitt, B.: Emotion and motivation: The role of the amygdala, ventral striatum, and prefrontal cortex. Neuroscience and Biobehavioral Reviews 26(3), 321–352 (2002)
14. Balkenius, C., Morén, J.: Computational models of classical conditioning: A comparative study. In: Proc. 5th Int. Conf. Simulation of Adaptive Behavior: From Animals to Animats, vol. 5, pp. 348–353 (1998)
15. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation. Data Mining and Knowledge Disc. 8, 53–87 (2004)
16. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytologist 11(2), 37–50 (1912)
17. Lipkus, A.: A proof of the triangle inequality for the Tanimoto distance. J. Mathematical Chemistry 26(1), 263–265 (1999)
18. Ravindran, B., Barto, A.: Approximate homomorphisms: A framework for non-exact minimization in Markov decision processes. In: Proc. 5th Int. Conf. Knowledge-Based Computer Systems (2004)
19. Givan, R., Dean, T., Greig, M.: Equivalence notions and model minimization in Markov Decision Processes. Artificial Intelligence 147, 163–223 (2003)
20. Szepesvári, C., Smart, W.: Interpolation-based Q-learning. In: Proc. 21st Int. Conf. Machine Learning, pp. 100–107 (2004)
21. Ormoneit, D., Sen, S.: Kernel-based reinforcement learning. Machine Learning 49, 161–178 (2002)