Under review for the 16th European Conference on Machine Learning (ECML-05)

Relevance-Weighted Action Selection in MDP's

Alexander A. Sherstov and Peter Stone

Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712 USA
{sherstov,pstone}@cs.utexas.edu

Abstract. Temporal-difference reinforcement learning (RL) has proven to be an effective approach to sequential decision making. A central issue in RL is action selection. We propose a novel action-selection method based on *action relevance*, a measure of the "optimality potential" of an action. We analyze its workings in an RL context and establish its effectiveness at accelerating learning. One challenge is that the learner initially has no knowledge of the relevances. We contribute a powerful and easy-to-use technique, *RTP transfer*, that estimates relevances on a task from past experience with related tasks, motivate it theoretically, and validate it empirically in the maze, cart control, and pendulum swing-up domains.

1 Introduction

Temporal-difference RL [1] has proven to be an effective approach to sequential decision making and is the subject of much current research. A central issue in RL is *action selection*. When using temporal-difference learning, the learner maintains estimates of the utility of each action in each state. Provided these estimates are accurate, the optimal behavior is simply to select the action with the highest estimate. However, this "greedy" course of action is not appropriate if the estimates are inaccurate, e.g., at initial stages of learning or if the environment is continually changing. Exploration is thus necessary to discover an optimal policy in a stationary environment or to adjust in a changing one.

Exploration is inherently *costly*. When the learner takes an exploratory action, it may be able to improve its policy if its prescribed action is suboptimal. However, it risks picking a worse action than its current choice and thus obtaining a worse return than it would through exploitation. Exploratory activity is therefore a limited, valuable resource that requires careful budgeting.

A popular method for trading off exploration and exploitation is ϵ-*greedy action selection* [1]. Compared with other techniques, ϵ-greedy has been thoroughly studied and understood and offers a particularly compelling balance of ease of use, scope of applicability, and performance. These factors have ensured a substantial role for ϵ-greedy selection in modern RL research. At each step, ϵ-greedy explores (picks an action at random) with probability ϵ, and exploits (selects the current best action) with probability $1 - \epsilon$. The ϵ-greedy method treats all actions *equally* for the purpose of exploration. While it fairs well on small tasks, challenging real-world domains favor more informed exploration. A key premise of this work is that exploration of actions should be proportional to their *optimality potential* in a given state, i.e., their predicted payoff.

As a measure of optimality potential, we adopt *action relevance*: $\text{RELEVANCE}(a) = |\{s \in \mathcal{S} : \pi^*(s) = a\}|/|\mathcal{S}|$, where $\mathcal{S}$ and π^* are the state set and an optimal policy for the given MDP. In words, the relevance of an action is the fraction of states at which it is optimal. Alternatively, the relevance of an action is the likelihood of its being optimal in a given state. We propose a modification to ϵ-greedy action selection that selects an exploratory action with probability equal to the action's relevance, a method we call *relevance-weighted action selection*.

Initially, the learner has no knowledge of the action relevances. In what follows, we develop a powerful and easy-to-use knowledge-transfer method for estimating action relevances on a task from past experience with related tasks in the domain. The method, *RTP transfer*, applies to a broad range of learning settings in which multiple related tasks are to be mastered. We present comparative experiments in the maze, cart control, and pendulum swing-up domains that substantiate the benefits of relevance-weighted action selection with the proposed relevance-estimation method.

Additionally, this paper contributes one of the few successful methodologies for *knowledge transfer* in MDP's. The importance of leveraging past experience to accelerate the mastery of new tasks has been widely recognized. While generalization *within* tasks is a well-understood and largely solved problem (typically using function approximation), knowledge transfer *between* tasks has seen limited progress to date (see "Related Work" below for some notable exceptions). We demonstrate that our knowledge-transfer approach to action selection is at once simple to use and empirically successful. The theoretical analysis in this paper shows that RTP transfer is also a formally motivated, principled method.

2 Background

A *Markov decision process* (MDP) is a quadruple $\langle \mathcal{S}, \mathcal{A}, \mathbf{t}, r \rangle$, where $\mathcal{S}$ is a set of *states*; $\mathcal{A}$ is a set of *actions*; $\mathbf{t} : \mathcal{S} \times \mathcal{A} \to \Pr(\mathcal{S})$ is a *transition function* indicating a probability distribution over the next states upon taking a given action in a given state; and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a *reward function* indicating the reward upon taking a given action in a given state. Given a sequence of rewards $r_0, r_1, \ldots, r_n$, the associated *return* is $\sum_{i=0}^{n} \gamma^i r_i$, where $0 \leq \gamma \leq 1$ is the *discount factor*. Given a *policy* $\pi : \mathcal{S} \to \mathcal{A}$ for acting, its *value function* $V^\pi : \mathcal{S} \to \mathbb{R}$ yields, for every state $s \in \mathcal{S}$, the expected return from starting in state s and following π. The objective is to find an *optimal policy* $\pi^* : \mathcal{S} \to \mathcal{A}$ whose value function dominates that of any other policy at every state.

The learner experiences the world as a sequence of states, actions, and rewards, with no prior knowledge of the functions $\mathbf{t}$ and r. A practical vehicle for learning in this setting is the *Q-value function* $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, defined as $Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbf{t}(s'|s,a) V^\pi(s')$. The widely used Q-learning algorithm [2] incrementally improves an approximation to the Q-value function of the optimal policy.

3 Estimating Relevance

In many domains, action relevances are similar across tasks. Consider, for example, a pastry chef experimenting with a new recipe. Several parameters, such as oven temperature and time to rise, need to be determined. But based on past experience, only a

small range of values is likely to be worth testing. Similarly, when driving, the same safety practices (gradual acceleration, minor adjustments to the wheel) apply regardless of the terrain, destination, or vehicle. Finally, a bidding agent in an auction can raise a winning bid by any amount. But past experience in a different auction may suggest that only a small number of raises are worth considering. This section presents two techniques, *relevance transfer* and *randomized task perturbation*, that exploit domain-wide knowledge acquired on one task to accelerate learning on all subsequent ones. To highlight the roles of the tasks in knowledge transfer, we refer to the first task learned as *auxiliary* and to any subsequent tasks, as *primary*.

3.1 Relevance Transfer

Relevance transfer is a novel relevance-estimation method that applies when a learner is presented with two or more *related* tasks with identical action sets, all of which must be learned. Since real-world problems are rarely handled in isolation, this setting is quite common. Relevance transfer solves the auxiliary task from scratch with uniform relevances (i.e., using traditional ϵ-greedy action selection), identifying a near-optimal policy π^*. The method computes the relevance of each action from π^* directly from the definition: $\text{RELEVANCE}(a) = |\{s \in \mathcal{S} : \pi^*(s) = a\}|/|\mathcal{S}|$.[1] Relevance transfer learns any subsequent tasks using these newly discovered relevances: an action is selected on an exploratory move at random with probability equal to its computed relevance. When action relevances are similar across tasks in a domain, relevance transfer accelerates the mastery of the second and any subsequent tasks. In particular, it provides the information necessary to perform more effective exploration.

3.2 Randomized Task Perturbation (RTP)

Relevance transfer relies on the similarity of the tasks involved; if the auxiliary task is not representative of the others, relevance transfer can handicap the learner. If *many* tasks are to be learned, a straightforward remedy is to transfer from *multiple* tasks, learning each from scratch with uniform relevances. However, in some cases the learner may not have access to a representative sample of tasks in the domain. Furthermore, the cost of learning multiple tasks independently could be prohibitive.

We therefore focus on the harder problem of identifying the approximate action relevances in the domain by learning only *one* task from scratch and tackling all subsequent tasks with the resulting relevances. We propose *randomized task perturbation*, an enhancement to relevance transfer that makes it robust to misleading auxiliary tasks.

Detrimental Transfer Direct relevance transfer can be successful in many cases. However, in order to motivate the power of RTP, we introduce here an example in which direct transfer is detrimental. Consider a deterministic 4×4 grid world. A single cell is distinguished as a goal, whose location varies from task to task. The goal is an absorbing state with reward 1; all other transitions yield zero reward. The discount factor is $\gamma = 0.9$. The initial state is picked at random, and the objective is to reach the goal via the shortest path. Over all possible tasks in this domain (distinguished by the goal location), the four actions NORTH, SOUTH, EAST, and WEST are equally relevant. Thus,

[1] In continuous-state domains, the state space can be discretized or sampled.

knowledge transfer cannot improve on the default, learning with all identical relevances. (See "Empirical Results" below for domains in which relevance trends *do* exist across tasks and knowledge transfer *is* helpful.)

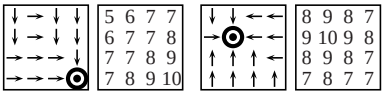

Fig. 1: An auxiliary and primary tasks.

Unfortunately, naïve transfer can hurt performance. Figure 1 features a pair of auxiliary and primary tasks, along with their optimal policies and value functions. The auxiliary task has its goal in the southeast corner of the map. The primary task is identical except the goal is moved to a northwesterly location. Although the auxiliary and primary tasks seem similar, transfer severely disadvantages the learner. The optimal policy for the auxiliary task, shown in Figure 1, includes only SOUTH and EAST actions. The primary task, on the other hand, features all four directions of travel in its optimal policy. The action relevances in the auxiliary task are completely unrepresentative of the primary task. Direct transfer predicts low relevances for the useful actions (NORTH and WEST) and vice versa and thus impedes learning.

Coping with Unrepresentative Past Experience RTP is a technique that permits a more thorough exposure to the domain's task space while learning in the same auxiliary task. The method works by internally distorting the optimal value function of the auxiliary task, thereby inducing an artificial *new* task while operating in the *same* environment. RTP transfer learns the optimal policy in this artificial task and computes the relevance of each action as the average over its relevance in the original and artificial tasks. Compared with direct transfer, RTP transfer is biased more toward a "generic" task in the domain and less toward the auxiliary task. If the auxiliary task is in fact more representative of the primary task than is a "generic" task in the domain, direct transfer may be preferable. By contrast, RTP strives for *safe* knowledge transfer, weighting domain-wide trends more than the idiosyncrasies of the auxiliary task. This technique is beneficial in the many domains that exhibit relevance trends across all tasks; a few are illustrated in "Empirical Results" below.

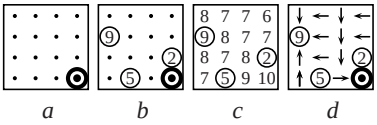

Fig. 2: RTP transfer at work.

Figure 2 illustrates the workings of RTP transfer. Figure 2*a* depicts the familiar auxiliary task. RTP distorts the task's optimal value function by randomly selecting a small fraction ϕ of the states and labeling them with randomly chosen values, drawn uniformly from $[v_{\min}, v_{\max}]$. Here $v_{\min} = r_{\min}/(1-\gamma)$ and $v_{\max} = r_{\max}/(1-\gamma)$ are the smallest and largest state values in the domain. We assume that the smallest and largest one-step rewards $r_{\min}$ and $r_{\max}$ can be readily estimated or learned.

The selected states form a set $\mathcal{F}$ of *fixed-valued states*. Figure 2*b* shows these states and their assigned values on a sample run with $\phi = 0.2$. RTP transfer learns the value function of the artificial task *by treating the values of the states in $\mathcal{F}$ as constant*, and by iteratively refining the other states' values via Q-learning. Figure 2*c* illustrates the resulting optimal values. The fixed-valued states have retained their assigned values, and the other states' values have been computed with regard to these fixed values.

RTP created an artificial task quite different from the original. The optimal policy in Figure 2*d* features *all four directions of travel*, despite the goal's southeast location.[2] As a result, the relevance estimates for the four actions are more in agreement with the domain-wide trend (uniform relevances, in this case) and less reflective of the peculiarities of the auxiliary task than with direct transfer. RTP has performed knowledge transfer gracefully with misleading experience.

RTP-TRANSFER($\phi, v_{\min}, v_{\max}$)

1 Add each $s \in \mathcal{S}$ to $\mathcal{F}$ with probability ϕ
2 **foreach** $s \in \mathcal{F}$
3 **do** $\mathit{random\text{-}value} \leftarrow \mathrm{rand}(v_{\min}, v_{\max})$
4 $Q^+(s,a) \leftarrow \mathit{random\text{-}value}$ for all $a \in \mathcal{A}$
5 **repeat** $s \leftarrow$ current state, $a \leftarrow \pi(s)$
6 Take action a, observe reward r, new state s'
7 $Q(s,a) \overset{\alpha}{\leftarrow} r + \gamma \max_{a' \in \mathcal{A}} Q(s',a')$
8 **if** $s \in \mathcal{S} \setminus \mathcal{F}$
then $Q^+(s,a) \overset{\alpha}{\leftarrow} r + \gamma \max_{a' \in \mathcal{A}} Q^+(s',a')$
9 **until** converged
10 $\forall a : \rho'(a) \leftarrow \dfrac{|\{s \in \mathcal{S} : a = \arg\max_{a' \in \mathcal{A}} Q(s,a')\}|}{|\mathcal{S}|}$,
$\rho''(a) \leftarrow \dfrac{|\{s \in \mathcal{S} \setminus \mathcal{F} : a = \arg\max_{a' \in \mathcal{A}} Q^+(s,a')\}|}{|\mathcal{S} \setminus \mathcal{F}|}$
11 **return** $\rho = \frac{1}{2}\rho' + \frac{1}{2}\rho''$

Fig. 3: RTP in pseudocode. The left arrow "$\leftarrow$" denotes assignment; $x \overset{\alpha}{\leftarrow} y$ denotes $x \leftarrow (1-\alpha)x + \alpha y$.

Figure 3 specifies RTP transfer in pseudocode, embedded in Q-learning. Q^+ denotes the Q-value function for the artificial task; $\rho'(a)$ and $\rho''(a)$ denote the learned relevance of an action in the auxiliary and artificial tasks, respectively.

Notes on RTP Transfer On a practical note, RTP transfer is easy to use. The algorithm's only parameter, ϕ, offers a tradeoff: $\phi \approx 0$ results in an artificial task almost identical to the original; $\phi \approx 1$ induces a value environment completely unrelated to the domain's transition and reward dynamics and thus unrepresentative of tasks in the domain. Importantly, RTP does not require any environmental interaction of its own—it reuses the $\langle s, a, r', s' \rangle$ quadruples generated while learning the unmodified auxiliary task. It may be useful to run RTP several times. A data-economical implementation learns all artificial Q-value functions Q_1^+, Q_2^+, etc., simultaneously within the same algorithm. The data requirement of this implementation is the same as in traditional Q-learning with ϵ-greedy action selection. The space and running time requirements are a modest multiple k of those in Q-learning, where k is the number of additional (artificial) Q-value functions learned. In case of multiple artificial tasks, the returned relevance of each action is averaged over the original and all artificial tasks. One direction for future work is to explore the benefits of weighting the original and artificial tasks *differently* in computing relevance. RTP can be viewed as a solution to the *masking effect* [3] in classical AI, a phenomenon when extrapolating from past experience yields a poorer solution than learning from scratch.

Extensions to Continuous Domains RTP transfer can be readily extended to continuous state spaces. In this case, the set $\mathcal{F}$ cannot be formed from individual states; instead,

[2] We ignore the action choices in $\mathcal{F}$ since those states are semantically absorbing.

$\mathcal{F}$ should encompass *regions* of the state space, each with a fixed value, whose aggregate area is a fraction ϕ of the entire state space. A practical implementation of RTP for continuous tasks in Section 5 uses *tile coding* [1], a popular function-approximation technique that discretizes the state space into regions (each with a value that is shared by all states in it) and generalizes updates in each region to nearby regions. The method can be readily adapted to ensure that fixed-valued regions retain their values (e.g., by resetting their values after every update).

3.3 Theoretical Motivation for RTP

As discussed above, action relevances on the auxiliary task may be unrepresentative of the domain as a whole. In such cases, relevance transfer may be harmful—a situation RTP is designed to avoid. This section analyzes the performance degradation due to learning with skewed relevances and shows how RTP remedies the situation.

Formally, let $\tilde{\mathcal{A}}^* = \{a \in \mathcal{A} : \pi^*(s) = a \text{ for some } s \in \mathcal{S}\}$ be the *optimal action set* of the auxiliary task, and let $\mathcal{A}^*$ be the true optimal action set of the *primary* task. In relevance transfer, the primary task is learned using the transferred action set $\tilde{\mathcal{A}}^*$ (i.e., all actions whose relevances are non-zero), in the hope that $\tilde{\mathcal{A}}^*$ is "similar" to $\mathcal{A}^*$. If $\mathcal{A}^* \not\subseteq \tilde{\mathcal{A}}^*$, the best policy $\tilde{\pi}^*$ achievable with the transferred action set in the primary task may be suboptimal. This section bounds the decrease in the highest attainable value of a state of the primary task due to the replacement of the full action set $\mathcal{A}$ with $\tilde{\mathcal{A}}^*$. For more a detailed exposition, see [4].

Formalism We start with a formalism that separates the commonalities and differences of tasks in the domain. Rather than specifying the effects of an action as a probability distribution $\Pr(\mathcal{S})$ over next *states*, we specify it as a probability distribution $\Pr(\mathcal{O})$ over *outcomes* $\mathcal{O}$. $\mathcal{O}$ is the set of "nature's choices," or deterministic actions under *nature's* control [5]. Corresponding to every action $a \in \mathcal{A}$ available to the *learner* is a probability distribution (possibly different in different states) over $\mathcal{O}$. When a is taken, nature "chooses" an outcome for execution according to that probability distribution.

Each state in a task is labeled with a *class* from among $\mathcal{C}$. An action's reward and transition dynamics are identical in all states of the same class. Formally, for all $a \in \mathcal{A}$ and $s_1, s_2 \in \mathcal{S}$, $\kappa(s_1) = \kappa(s_2) \implies r(s_1, a) = r(s_2, a), \mathbf{t}(s_1, a) = \mathbf{t}(s_2, a)$, where $\kappa(\cdot)$ denotes the class of a state. Classes allow the definition of $\mathbf{t}$ and r as functions over $\mathcal{C} \times \mathcal{A}$, a set common to all tasks, rather than the task-specific set $\mathcal{S} \times \mathcal{A}$. Combining classes with outcomes enables a task-independent description of the transition and reward dynamics: $\mathbf{t} : \mathcal{C} \times \mathcal{A} \to \Pr(\mathcal{O})$ and $r : \mathcal{C} \times \mathcal{A} \to \mathbb{R}$.

A *task* within a domain is fully specified by its state set $\mathcal{S}$, a mapping $\kappa : \mathcal{S} \to \mathcal{C}$ from its states to the classes, and a specification $\eta : \mathcal{S} \times \mathcal{O} \to \mathcal{S}$ of the next state given the current state and an outcome. Figure 4 illustrates the complete formalism. Formally, domains and tasks are defined as follows:

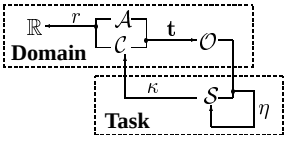

Fig. 4: Related-task MDP formalism.

Definition 1. *A domain is a quintuple* $\langle \mathcal{A}, \mathcal{C}, \mathcal{O}, \mathbf{t}, r \rangle$*, where* $\mathcal{A}$ *is a set of actions;* $\mathcal{C}$ *is a set of state classes;* $\mathcal{O}$ *is a set of action outcomes;* $\mathbf{t} : \mathcal{C} \times \mathcal{A} \to \Pr(\mathcal{O})$ *is a transition function; and* $r : \mathcal{C} \times \mathcal{A} \to \mathbb{R}$ *is a reward function.*

Definition 2. *A task within the domain $\langle \mathcal{A}, \mathcal{C}, \mathcal{O}, \mathbf{t}, r \rangle$ is a triple $\langle \mathcal{S}, \kappa, \eta \rangle$, where $\mathcal{S}$ is a set of states; $\kappa : \mathcal{S} \to \mathcal{C}$ is a state classification function; and $\eta : \mathcal{S} \times \mathcal{O} \to \mathcal{S}$ is a next-state function.*

This formalism captures implicit regularities in a typical domain: a fixed set of outcomes upon taking an action and large areas of the state space with identical dynamics. In a continuous setting, it is helpful to think of a given outcome $o \in \mathcal{O}$ as grouping similar (rather than identical) action effects; likewise for a class $c \in \mathcal{C}$.

Transfer Suboptimality Bound Given a fixed state s, let s_i denote the state that results if the ith outcome occurs. Then $\mathbf{v} = [V^*(s_1) \quad V^*(s_2) \quad \ldots \quad V^*(s_{|\mathcal{O}|})]^{\mathrm{T}}$ is the *outcome value vector* (OVV) for s, fully determined by the task. The set $U = \langle U_{c_1}, U_{c_2}, \ldots, U_{c_{|\mathcal{C}|}} \rangle$ of all OVV's of a task, grouped by the classes of their corresponding states, fully determines the optimal action set. OVV sets provide a natural measure of task dissimilarity $\Delta(U, \tilde{U}) \stackrel{def}{=} \max_{c \in \mathcal{C}} \max_{\mathbf{u} \in U_c} \left\{ \min_{\tilde{\mathbf{u}} \in \tilde{U}_c} ||\mathbf{u} - \tilde{\mathbf{u}}||_2 \right\}$: the worst-case distance between an OVV in the primary task (U) and the nearest OVV of the same class in the auxiliary task ($\tilde{U}$). This yields the desired bound (see [4] for a proof):

Theorem 1. *Let $\tilde{\mathcal{A}}^*$ be the optimal action set of the auxiliary task. Replacing the full action set $\mathcal{A}$ with $\tilde{\mathcal{A}}^*$ reduces the highest attainable value of a state in the primary task by at most $\Delta(U, \tilde{U}) \cdot \sqrt{2}\gamma/(1-\gamma)$, where U and $\tilde{U}$ are the OVV sets of the primary and auxiliary tasks, respectively.*

Analysis of RTP Theorem 1 implies that learning with the transferred relevances is safe if every OVV in the primary task has in its vicinity an OVV of the same class in the auxiliary task. To do well with unrepresentative auxiliary experience, the learner must sample the domain's OVV space not reflected in the auxiliary task. *Randomized task perturbation* (RTP) allows for a more thorough exposure to the domain's OVV space while learning in the same auxiliary task. In terms of the analysis above, the combined (original + artificial) OVV set in RTP is closer to, or at least no farther from, the primary task's OVV set than is the OVV set of the original auxiliary task alone. The algorithm thereby reduces the dissimilarity of the two tasks and improves the suboptimality guarantees of Theorem 1. A major strength of RTP transfer is its ability to extract relevance information about actions implicit in the domain-wide dynamics, given access to a single task and no model knowledge.

4 Testbed Domains

This section describes three common RL benchmarks that we used to assess the comparative efficacy of our proposed action-selection method. Two of the domains have continuous states; the remaining domain has discrete states.

4.1 Cart Control

Cart control (Figure 5) features a cart capable of moving left or right subject to the application of linear acceleration. The state of the cart is its position p and

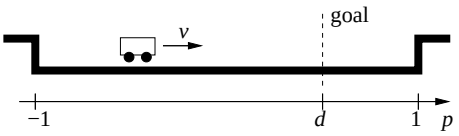

Fig. 5: Cart control domain.

velocity v. An action is an acceleration a. Initially, the cart is placed at a random location $p \in [-1, 1]$ and launched at a random speed $v \in [-1, 1]$. The objective is to drive the cart as quickly as possible to a destination $d \in [-1, 1]$ while using up as little power as possible. The dynamics of the cart are $p_{t+1} = p_t + v_t \Delta t$ and $v_{t+1} = v_t + a_t \Delta t$, where $\Delta t = 0.2$ is the simulated time step. When the next state is computed, its components (p and v) are additionally truncated if necessary to lie in $[-1, 1]$. The reward function is $r(p, v, a) = -(d - p)^2 - a^2$, and the discount factor is $\gamma = 0.95$. The optimal policy is to first accelerate in the direction of the goal, then to brake gradually as the cart approaches the goal, and finally to stop in the goal's immediate vicinity. Different tasks within this domain are distinguished by the location d of the goal. The action space is uniformly discretized to 11 values: $\{-1, -0.8, \ldots, 0, \ldots, 0.8, 1\}$. This domain is commonly known in the control literature as "double integrator."

4.2 Pendulum Swing-up

In the pendulum swing-up task (Figure 6), the objective is to swing a pendulum to an upright position by applying angular acceleration. The pendulum's state comprises angular position θ and angular velocity ω. Position θ ranges from 0 to 1 in the left semi-circle and from 0 to -1 in the right one; the downward vertical is simultaneously $\theta = 1$ and $\theta = -1$. An action is an angular acceleration α. Initially, the pendulum is launched at a random position and at a random velocity. As in the cart control task, the three variables range in $[-1, 1]$, the reward function is $r(\theta, \omega, \alpha) = -\theta^2 - \alpha^2$, and $\gamma = 0.95$. The pendulum's dynamics are $\theta_{t+1} = \theta_t + \omega_t \Delta t$ and $\omega_{t+1} = \omega_t + [\alpha + g \sin(\pi\theta)]\Delta t$, where $\Delta t = 0.2$ is the simulated time step. As the angular velocity equation indicates, the pendulum's movement is subject to gravity, measured as a multiple g of the maximum angular acceleration. At $g \geq 2$, it is impossible to swing the pendulum to an upright position without first rocking it in the opposite direction to gain momentum. Tasks in this domain are distinguished by gravity $g \geq 0$. The action set is uniformly discretized to $\{1, -0.8, \ldots, 0, \ldots, 0.8, 1\}$.

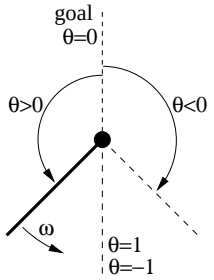

Fig. 6: Pendulum swing-up domain.

4.3 Maze

The 100×100 deterministic maze in Figure 7 features four actions: NORTH, SOUTH, EAST, and WEST. The states are the maze cells. Tasks in this domain are distinguished by the location of the goal, an absorbing state with reward 1; all other transitions yield zero reward. Moves into walls and outside the maze result in no change of cell. The objective is to reach the goal via the shortest path from the start cell, chosen each time at random. The discount rate is $\gamma = 0.95$.

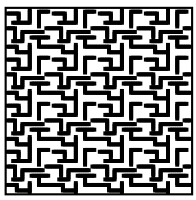
Fig. 7: Maze domain.

5 Empirical Results

Experimental Setup We used Q-learning with step size $\alpha = 0.1$ and $\epsilon = 0.2$ to compare relevance-weighted action selection to the traditional ϵ-greedy method in the

three domains above: cart control (goal position $d = 0$), pendulum swing-up (gravity $g = 2$), and maze (goal cell in the bottom-right corner). In each case, an episode started in a random state and continued for 200 time steps. The performance metric was average state value under the current greedy policy. The metric was computed by performing a large number of rollouts (state trajectories) at randomly chosen states and averaging the associated γ-discounted returns.

To estimate action relevances for use in the relevance-weighted method, we used direct transfer and RTP transfer. The parameters in RTP transfer were: $\phi = 0.20$, 1 trial (picked heuristically and not optimized). The transfer methods estimated relevances from an auxiliary task in the corresponding domain: cart control (goal at $d = -1$), pendulum swing-up (zero gravity), and maze (goal in the top-left corner). In cart control, the motion equation was additionally changed to $p_{t+1} = p_t + 2v_t\Delta t$, i.e., the effective speed was doubled when computing the next position, to simulate a slippery surface. These auxiliary tasks are quite different from the primary tasks, a design meant to put knowledge transfer to the test in a maximally challenging setting.

To learn in the continuous-state cart control and pendulum swing-up domains, we used *tile coding* [1]. The tile width was 0.008 for position (p, θ) and 0.04 for speed (v, ω). The finer resolution along the position dimension reflects its greater importance in decision making. We did not experiment with varying these parameters. In RTP, the updates in tile coding were followed by resetting the fixed values, as described earlier. The initialization was to the lowest value in each domain (-40, -40, and -20 in cart control, pendulum swing-up, and maze, respectively).

Relevance Estimates Figure 8 shows the true relevances on the three test tasks and their estimates obtained via knowledge transfer. The estimates obtained by direct transfer closely reflect features of the auxiliary tasks and thus offer a poor approximation to the true relevances. In particular, the auxiliary cart-control task induced negligible relevances for rightward accelerations (due to the left-hand location of the goal); the true relevances are balanced for rightward and leftward accelerations. The zero-gravity auxiliary pendulum swing-up task made large accelerations seem unrealistically useful because without gravity, the pendulum can be swung up to vertical by a single forceful push rather than by gradually rocking it back and forth to gain momentum. Likewise, the relevances on the maze task are skewed by the unrepresentative top-left location of the goal in the auxiliary task.

By contrast, RTP transfer yielded relevance estimates more representative of the domain and thus closer to the test tasks. In particular, the relevances in cart control and

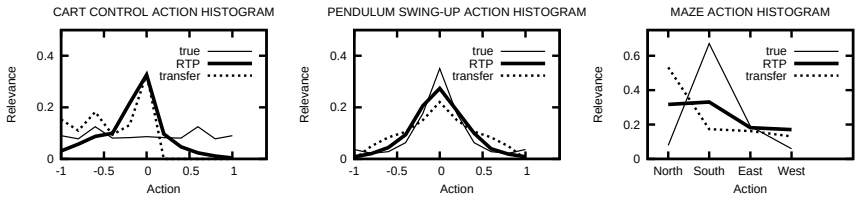

Fig. 8: Action relevances estimated via knowledge transfer.

pendulum swing-up are symmetric for leftward and rightward accelerations (reflecting the goal's expected placement at the center) and are lower for large accelerations (reflecting their high cost). There is no such domain-wide bias in the maze domain, resulting in nearly uniform relevances. In the event that the auxiliary task is in fact more representative of the primary task than is a "generic" task in the domain, direct relevance transfer might be preferable. On the other hand, RTP transfer provides for *safe* (if occasionally suboptimal) knowledge transfer *regardless* of the auxiliary task.

Results Figure 9 plots performance on the test task in the three domains using relevance-weighted action selection via the two relevance-estimation methods. Traditional ϵ-greedy action selection, as well as relevance-weighted action selection with the true relevances, are plotted as baselines. The true-relevance baseline is included for analysis purposes only and does *not* represent an actual method (since the true relevances are unknown ahead of time). The time scale of Figure 9, 1 million episodes, arguably exceeds any realistic allotment of training time. The meaning of the curve labels is as follows:

- TRUE: Relevance-weighted selection with the *true* relevances on the test task.
- RTP: Relevance-weighted selection with relevances *estimated* from an auxiliary task via RTP transfer.
- TRANSFER: Relevance-weighted selection with relevances *estimated* from an auxiliary task via direct transfer.
- UNIFORM: Traditional ϵ-greedy selection. (Equivalent to relevance-weighted selection with uniform relevances.)

In all cases, relevance-weighted action selection with true relevances yields the best performance for the first 1 million episodes. In the maze domain, this method is *eventually* overtaken by uniform (traditional ϵ-greedy) action selection. This is because relevance is but an *estimate* of an action's optimality potential, and the optimal action in a given state is not always assigned the highest relevance. In fact, the optimal action will occasionally have relevance below $1/|\mathcal{A}|$ (uniform). As a result, relevance-weighted selection neglects optimal actions at certain states, unlike the uniform method. Given enough training time, traditional ϵ-greedy action selection may thus achieve a better policy. The near-identical performance of the two methods on the cart control task is due to a different reason: the true action relevances are nearly uniform.

The performance of direct relevance transfer is not particularly positive: the method is beneficial on the pendulum swing-up task but substantially worse than traditional

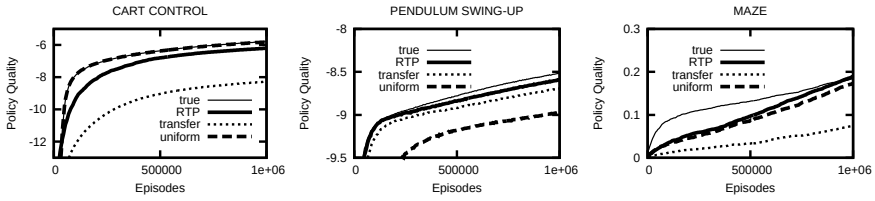

Fig. 9: Performance on the maze, cart control, and pendulum swing-up tasks. Each curve is a point-wise average over 100 runs. All mentioned comparisons are statistically significant at 0.01.

ϵ-greedy selection on the other two tasks. RTP transfer is always superior to direct transfer. RTP improves on ϵ-greedy on the maze task, confidently outperforms it on pendulum swing-up (and comes a close second to the true-relevance benchmark!), and slightly under-performs ϵ-greedy on cart control. RTP thus exploits domain-wide similarities where they exist and is minimally distracted by the unrepresentative peculiarities of the auxiliary task. These results reflect the difficulty of transferring knowledge and the effectiveness of RTP transfer at this task.

6 Related Work

Action selection has long been a vital research problem in RL. Several action-selection methods other than ϵ-greedy have been developed, including softmax selection [1], interval estimation [6], and E^3 [7]. Whereas these other methods focus on information about actions tried and value estimates on a single task during learning, this paper proposes a method for leveraging information from an auxiliary task to enable more informed exploration. An avenue for future work is to blend relevance-weighted action selection with these other methods so as to initially exploit knowledge from an auxiliary task and then gradually make use of value estimates from the primary task.

An alternate characterization of this work is the use of knowledge transfer to accelerate learning. Knowledge transfer has been obtained on hierarchical [8, 9], first-order [5], and factored [10] MDP's. A limitation of this related research is the reliance on a human designer for an explicit description of the regularities in the domain's dynamics, be it in the form of matching state regions in two problems, a hierarchical policy graph, relational structure, or situation-calculus fluents and operators. By contrast, RTP transfer discovers and exploits similarities across tasks in the domain to the extent that they are present and requires no human guidance along the way. Furthermore, our method is robust to unrepresentative auxiliary experience.

In some sense, RTP transfer can be seen as the counterpart to learning search control knowledge in classical AI planning [11]. Though the mechanism by which actions are selected is entirely different in RL, information from a previous task in the same domain is used to prune the space of possible actions. It is in this context that the masking effect, mentioned above in the discussion of RTP, has been studied previously [3].

7 Conclusions and Future Work

This paper introduces *action relevance*, defined as the fraction of states at which an action is optimal. Relevance is a valuable predictor of an action's optimality potential in a state. We demonstrate how action relevance can be exploited to substantially accelerate learning. However, the learner initially has no knowledge of relevances. We introduced a powerful knowledge-transfer technique, *RTP transfer*, that automatically leverages similarities across tasks in a domain to estimate action relevances on a given task from past experience on related tasks in the domain. The method is simple to use and applies to a broad range of learning settings in which *multiple* related tasks must be mastered; it performs well even with misleading past experience. This paper documents a successful application of RTP transfer to the continuous-state cart control and pendulum swing-up

domains, as well as a discrete maze domain. The theoretical analysis above additionally shows that RTP is a principled, formally motivated approach.

A promising direction for future work is to enhance the predictive power of action relevance by computing it over a *restricted* region of the state space. In many domains, not all states are worth considering in computing an action's relevance: some are rarely or never encountered, either in general or under optimal behavior. Removing those states from consideration would allow the learner to focus on the more useful states. Ideally, relevance would reflect differences in an action's optimality potential at different states, rather than offering a single estimate for the entire state space. The possibility of obtaining such estimates via knowledge transfer merits further research. More generally, it is intriguing to explore other definitions of relevance. The one used in this paper (fraction of states at which the action is optimal) is not the only possibility; any quantity that generalizes well from task to task and does not require mapping states between tasks will do. An orthogonal line of work is to adjust action relevance over time to reflect the evolving Q-values.

8 Acknowledgments

The authors are thankful to Lilyana Mihalkova for helpful discussions and her feedback on earlier versions of this manuscript. This research was supported in part by NSF CAREER award IIS-0237699, DARPA grant HR0011-04-1-0035, and an MCD fellowship.

References

1. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
2. Watkins, C.J.C.H.: Learning from Delayed Rewards. PhD thesis, Cambridge University (1989)
3. Tambe, M., Rosenbloom, P.S.: On the masking effect. In: AAAI. (1993) 526–533
4. Sherstov, A.A., Stone, P.: Improving action selection in MDP's via knowledge transfer. In: Proc. 20th National Conference on Artificial Intelligence (AAAI-05), Pittsburgh, USA (July 9–13, 2005) To appear.
5. Boutilier, C., Reiter, R., Price, B.: Symbolic dynamic programming for first-order MDPs. In: Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Seattle, WA (2001) 690–697
6. Kaelbling, L.P.: Learning in embedded systems. MIT Press (1993)
7. Kearns, M.J., Singh, S.P.: Near-optimal reinforcement learning in polynominal time. In: Proc. of the Fifteenth International Conference on Machine Learning (ICML-98), Morgan Kaufmann Publishers Inc. (1998) 260–268
8. Hauskrecht, M., Meuleau, N., Kaelbling, L.P., Dean, T., Boutilier, C.: Hierarchical solution of Markov decision processes using macro-actions. In: Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98). (1998) 220–229
9. Dietterich, T.G.: Hierarchical reinforcement learning with the MAXQ value function decomposition. Journal of Artificial Intelligence Research **13** (2000) 227–303
10. Guestrin, C., Koller, D., Gearhart, C., Kanodia, N.: Generalizing plans to new environments in relational MDPs. In: Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI-03). (2003)
11. Minton, S.: Learning Search Control Knowledge: An Explanation-Based Approach. Kluwer Academic Publishers (1988)