Neural Process Lett
DOI 10.1007/s11063-012-9248-7

Towards a Self-Learning Agent: Using Ranking Functions as a Belief Representation in Reinforcement Learning

Klaus Häming · Gabriele Peters

Abstract We propose a combination of belief revision and reinforcement learning which leads to a self-learning agent. The agent shows six qualities we deem necessary for a successful and adaptive learner. This is achieved by representing the agent's belief in two different levels, one numerical and one symbolical. While the former is implemented using basic reinforcement learning techniques, the latter is represented by Spohn's ranking functions. To make these ranking functions fit into a reinforcement learning framework, we studied the revision process and identified key weaknesses of the to-date approach. Despite the fact that the revision was modeled to support frequent updates, we propose and justify an alternative revision which leads to more plausible results. We show in an example application the benefits of the new approach, including faster learning and the extraction of learned rules.

Keywords Hybrid learning system · Belief revision · Ranking functions · Reinforcement learning

1 Introduction

This paper discusses an approach towards a system which represents a self-learning agent that exhibits a number of important capabilities [14]. The list of these capabilities includes perception, recognition, reasoning, planning, decision making, and goal oriented behavior to name a few. There are six properties which in our opinion are necessary for an agent that shows the aforementioned capabilities. In a nutshell, these are

1. hierarchical learning, to regard the differences between implicit and explicit learning
2. emerging mechanisms, especially the creation of symbolic representations from a numerical representation

K. Häming (✉) · G. Peters
University of Hagen, Universitätsstr. 1, 58097 Hagen, Germany
e-mail: klaus.haeming@fernuni-hagen.de

G. Peters
e-mail: gabriele.peters@fernuni-hagen.de

Published online: 28 September 2012

3. multi-directional transfer, i.e., the exchange of information between the learning levels
4. generalization, understood as the generation of abstractions from example
5. exploration, to allow the acquisition of belief from scratch
6. adaptivity, meaning the application of non-stationary belief models

Some of these properties are already captured in reinforcement learning [22], an often applied approach in learning from a series of perceptions. These properties are exploration, adaptivity and, to some degree, emergent behavior, since a reinforcement learning agent has to learn from experience alone while constantly updating a belief representation.

Because in its usual incarnation, reinforcement learning only numerically captures the expected reward of available state transitions, it is practically impossible to allow further reasoning on these numbers directly, neither for the agent, nor for a human being. We aim to improve this and also include the other three properties by augmenting the reinforcement learning agent with a second learning level. This is inspired by psychological findings [2,8,15,20] which indicate that such a two-level learning architecture can explain some of the human learning abilities. An earlier instance of such a two-level architecture is the CLARION model [19] which encodes both, the numerical and the symbolical level in neural networks.

However, to model the second learning level we adopt the more "natural" choice of employing a belief revision technique. Belief revision represents an agent's belief symbolically and hence facilitates further reasoning. Because the agent repeatedly revises its belief, we chose Spohn's ranking functions [12,18], which are designed to allow this. This combination has been proposed before [13], influenced by [19,24], but this paper points out a major flaw in the previously used revision, proposes a new one, and provides evidence for its superiority.

For the lower learning level, we use a basic Q-learning scheme. Since this work focuses more on the higher learning level, we describe this only briefly. Also, while humans are able to learn top-down or bottom-up [21], we focus on the bottom-up part, while the top-down part is implemented rather implicitly though a restriction of the actions the agent may choose from.

This work is also related to the topic of relational reinforcement learning [6]. Since it is mainly concerned with representing relations in the Q-function, it also needs to include information which is naturally described in symbolic form. Therefore, a number of approaches such as defining a distance measure on the relations and applying a nearest-neighbor interpolation [5], using decision trees as Q-functions [3] or applying kernel methods have been proposed [7].

While we have already described the general idea of our approach briefly in [9], we present here the detailed formalism, extended examples, and give a theoretical justification.

2 Notes on Notation

This section briefly addresses the notations that will be used in later sections.

A variable a can represent a value from its domain $\mathfrak{D}_a$. Such a domain consists of discrete values. One such realization of a variable is called a *literal*. We write literals by denoting the variable as a subscript of its value (e.g., 3_a or $\mathtt{t}_a$). A *formula* consists of literals and logical operators such as $\wedge$, $\vee$, $\Rightarrow$, etc. It is referred to by an uppercase letter, e.g., $A := 0_a \vee 1_b$. A *negation* of a literal refers to a formula. For example, if $\mathfrak{D}_a := \{1, 2, 3\}$, then

$$\overline{2_a} = (1_a \vee 3_a) \tag{1}$$

and

$$\text{and} \quad 1_b \wedge \overline{2_a} \wedge 7_c \Leftrightarrow 1_b \wedge (1_a \vee 3_a) \wedge 7_c. \tag{2}$$

By convention, the "$\wedge$"-operator may be omitted.

The set of all variables is $\mathfrak{V}$, while the set of variables that are realized in a formula A is denoted by $\mathfrak{V}_A$.

A *model* is a conjunction in which exactly one literal exists for each variable. The set of all models is referred to as $\mathfrak{M}$. If we restrict the set of variables the models are derived from, we write the variable set as a subscript, e.g. $\mathfrak{M}_{\mathfrak{V}}$. Consequently, the set of models that are derived from the variables present in a specific formula A is denoted by $\mathfrak{M}_{\mathfrak{V}_A}$. A model M is said to be *a model of a formula* F, if F is true for the literals in M. We denote this as $M \models F$. If an agent believes in a formula A, which means A can be inferred from its belief base κ, we write $\kappa \models A$.

A conditional is denoted by $A \Rightarrow B$, where A is the antecedent and B is the consequent. The set of conditionals we obtain when the antecedent A is replaced by a set of formulas $\mathfrak{F}$, is referred to as $\{\mathfrak{F} \Rightarrow B\} := \{F \Rightarrow B | F \in \mathfrak{F}\}$.

3 Linking Reinforcement Learning and Belief Revision

Let us assume an *environment* that is described as a *set of states*. State transitions are performed depending on the current state and the current *action* carried out by the agent. The transitions are rewarded. A goal of reinforcement learning consists in the identification of beneficial actions, i.e., those actions that produce high rewards. More concisely, we have:

- A set of states $\mathfrak{S}$
- A set of actions $\mathfrak{A}$
- A transition function $\delta : \mathfrak{S} \times \mathfrak{A} \rightarrow \mathfrak{S}$
- A reward function $r : \mathfrak{S} \times \mathfrak{A} \rightarrow \mathbb{R}$

Belief about good and poor actions is established by applying a learning technique. In our approach we apply Q-learning. This technique has the convenient property of being *policy*-free. This means that the result does not depend on the strategy with which the agent explores the environment.

The agent's experience is captured in the Q(uality)-function that assigns an expected reward to each state-action-pair. The Q-function is updated after each state transition in the following way:

$$Q(S, A) = r + \gamma \max_{A'} Q(S', A') \tag{3}$$

with

$$S' := \delta(S, A') \tag{4}$$

One can interpret this formula in the way that the agent will *believe* an action A to be a best action, if it has the highest $Q(S, A)$ value for a given state S. This is the point where we establish a connection to the high-level belief using belief revision in the following. Belief revision is a theory of maintaining a belief base in such a way that the current belief is represented in a consistent manner [1,4]. We model our belief base κ as an instance of Spohn's ranking function [17]. Such a ranking function maintains a list of all models. The models the agent believes in are set to rank 0, while all ranks greater than 0 represent an increasing disbelief. We denote the rank a ranking function κ assigns to a model M as $\kappa(M)$. By convention, contradictions shall have the rank ∞. The operator "$\models$" of Sect. 2 is defined for a ranking function as

$$\begin{aligned} \kappa \models A :\Leftrightarrow\ & (\exists M_1, M_1 \models A : \kappa(M_1) = 0) \\ & \wedge\ (\forall M_2, M_2 \models \overline{A} : \kappa(M_2) > 0) \end{aligned} \tag{5}$$

which requires a believed formula to have a model with rank 0 and its negation to have a rank greater than 0.

In this work, the states and actions are described as formulas. Therefore it is possible to store information on them in a suitable ranking function. For instance, let the state description consist of n variables. Then, a complete state description has the form

$$S = s_1 \wedge s_2 \wedge \cdots \wedge s_n,$$

where each variable takes a value of its domain. An action is described by a single variable:

$$A = a$$

A formula composed of a conjunction of the state description and the action, such as

$$M = s_1 \wedge s_2 \wedge \cdots \wedge s_n \wedge a,$$

captures naturally the information of any state-action-pair. Since there are no other variables, this formula is a model. Our ranking function κ comprises exactly this kind of models. The interplay between the ranking functions and the Q-function is described in the context of an example application in Sect. 8.

4 The Revision of Ranking Functions

The current belief represented by the ranking functions consists of models, i.e., propositional information in the form of conjunctions. However, during exploration the information gathered and the information needed is in the form of conditionals. So, in a particular state S we need to know if $S \Rightarrow A$, not SA.

To check, whether a ranking function believes in a conditional the agent can temporarily believe its antecedent (known as *conditioning*) and check if the conjunction of the antecedent and the consequent is also believed. At the same time, the conjunction of the antecedent and the negation of the consequent must not be believed (that is, $\kappa(S\overline{A}) > 0$). Generally, we do not have to condition κ to find out whether a conditional is believed. It is sufficient to compute the belief ranks $r_1 = \kappa(SA)$ and $r_2 = \kappa(S\overline{A})$. If $r_1 < r_2$, the conditional will be believed. This comparison of ranks is done whenever the agent needs to decide what action to take.

More difficult than querying the belief base is its update, called *revision*. The revision operator is "$*$". Conditionals in belief revision are usually denoted by $(A|S)$, where S is the antecedent and A the consequent. The meaning of $(A|S)$ is not exactly the same as $S \Rightarrow A$ [12]. The latter means that S implies A irrespective of the values of other variables. In contrast, $(A|S)$ expresses that A will be believed if κ is conditioned with S and S alone, therefore a revision $(\kappa * (ST))$ may not result in A being believed.

In our context of reinforcement learning, if S is a complete state description, it will capture all the available information. Then, an expression such as $ST, T \neq S$ is necessarily a contradiction and therefore not believed. In this case, the meaning of $S \Rightarrow A$ and $(A|S)$ is the same. Therefore, on a first attempt, we use $(\kappa * (A|S))$ to revise κ with a conditional analogous to [13]. Then, we examine the consequences of such a decision.

After a revision of κ with the conditional $S \Rightarrow A$, we want

$$(\kappa * (A|S))(SA) < (\kappa * (A|S))(S\overline{A}) \tag{6}$$

to hold. If this is already the case, nothing has to be done. Otherwise the following holds:

Theorem 1 *If $\kappa(SA) \geq \kappa(S\overline{A})$, then the ranking function κ' derived from κ by rearranging the models using*

$$\begin{aligned} \forall M \in \mathfrak{M} : \kappa'(M) &:= (\kappa * (A|S))(M) \\ &= \begin{cases} \kappa(M) - \kappa(S \Rightarrow A) & : M \models S \Rightarrow A \\ a + b & : M \models S\overline{A} \end{cases} \end{aligned} \tag{7}$$

with

$$\begin{aligned} a &= \kappa(SA) - \kappa(S \Rightarrow A) + 1 \\ b &= \kappa(M) - \kappa(S\overline{A}) \end{aligned}$$

will result in $\kappa'(SA) < \kappa'(S\overline{A})$. Consequently, κ' expresses the belief in $S \Rightarrow A$.

Proof Let us partition the models in κ into three disjoint sets:

$$\begin{aligned} \mathfrak{M}_1 &= \{M|M \models \overline{S}\}, \\ \mathfrak{M}_2 &= \{M|M \models SA\}, \text{ and} \\ \mathfrak{M}_3 &= \{M|M \models S\overline{A}\}. \end{aligned}$$

We address the first rule of Eq. 7 first. The purpose of it is to let $\kappa'(S \Rightarrow A) = 0$. The models in $\mathfrak{M}_1 \cup \mathfrak{M}_2$ are those that model $S \Rightarrow A$. Therefore we reduce in rank all models in $\mathfrak{M}_1 \cup \mathfrak{M}_2$ by $\kappa(S \Rightarrow A)$ which is the rank of the lowest ranked model in $\mathfrak{M}_1 \cup \mathfrak{M}_2$. Hence, $\kappa'(S \Rightarrow A) = 0$. We now consider term a of the second rule. We want $\kappa'(SA) < \kappa'(S\overline{A})$ to hold. That means, after revision, the lowest rank of the models in $\mathfrak{M}_3$ needs to be at least $\kappa'(SA)+1$. Since the models of SA are found in $\mathfrak{M}_2$ and are therefore shifted by the first rule, $\kappa'(SA) = \kappa(SA) - \kappa(S \Rightarrow A)$. Adding 1 is arbitrary but sufficient to meet the requirements. Term a alone would shift the ranks of all models of $\mathfrak{M}_3$ to the rank $\kappa'(SA) + 1$. To preserve the relative ranking of the models, we need to add term b to the second rule. Since $\kappa(S\overline{A})$ is the rank of the lowest ranked model of $\mathfrak{M}_3$, this very model is still shifted to the rank $\kappa'(SA) + 1$. The other models, however, now keep their distance. □

5 Negated Consequents

What will happen if κ is revised with $S \Rightarrow \overline{A}$? Then, an application of Eq. 7 will result in $(\kappa * (\overline{A}|S))(S\overline{A}) < (\kappa * (\overline{A}|S))(SA)$.

This does not mean that all models of $S\overline{A}$ have a rank lower than $\kappa(SA)$. We show this in the following example. Let us define two variables a and b with their domains $\mathfrak{D}_a := \{1, 2\}$ and $\mathfrak{D}_b := \{1, 2, 3\}$. The current belief is represented by a ranking function, where the first entry represents the current belief; that means its model has rank 0. Now, we want the following ranking function κ_{neg} to believe $1_a \Rightarrow \overline{1_b}$:

$$\kappa_{neg} = \left\| \begin{array}{l} 21 \\ \hline 11 \\ 22 \\ 12 \\ 23 \\ 13 \end{array} \right\| \xrightarrow{(\kappa_{neg} * (\overline{1_b}|1_a))} \kappa'_{neg} = \left\| \begin{array}{l} 21 \\ \hline \\ 22 \\ 12 \\ 11\ 23 \\ 13 \end{array} \right\| \tag{8}$$

which beliefs $1_a \Rightarrow 2_b$, but not $1_a \Rightarrow 3_b$. This behavior is perfectly sane since $(1_a \wedge 2_b) \wedge (1_a \wedge 3_b)$ is a contradiction. But the belief in $(1_a \wedge 1_b)$ is stronger than the belief in $(1_a \wedge 3_b)$. If we revise κ with $1_a \Rightarrow \overline{2_b}$, then the result will be

$$\kappa''_{neg} = (\kappa'_{neg} * (\overline{2_b}|1_a)) = \left\| \begin{array}{l} 21 \\ \hline \\ 22 \\ \\ 11\ 23 \\ 13\ 12 \end{array} \right\| . \tag{9}$$

This expresses a belief in $1_a \Rightarrow 1_b$ which is certainly not what we expect an agent to believe if it has just been exposed to the information $1_a \Rightarrow \overline{1_b}$ and $1_a \Rightarrow \overline{2_b}$. Instead, a belief in $1_a \Rightarrow 3_b$ seems reasonable.

6 Generalization

We examine a revision by Eq. 7 in the context of generalization by examining what effect the omission of variables in a formula has. Let us partition the set of variables $\mathfrak{V}$ into three *non-empty* subsets:

$$\begin{aligned} \mathfrak{V} &= \mathfrak{X} \cup \mathfrak{Y} \cup \mathfrak{Z}, \text{ with} \\ \mathfrak{X} \cap \mathfrak{Y} &= \emptyset, \mathfrak{X} \cap \mathfrak{Z} = \emptyset, \text{ and } \mathfrak{Y} \cap \mathfrak{Z} = \emptyset \end{aligned} \tag{10}$$

Next, take a model from each of the subsets, such as

$$\begin{aligned} X &\in \mathfrak{M}_{\mathfrak{X}} \\ Y &\in \mathfrak{M}_{\mathfrak{Y}} \\ Z &\in \mathfrak{M}_{\mathfrak{Z}} \end{aligned} \tag{11}$$

The revision $\kappa * (Z|X)$ will lead to a belief base that believes a particular model M' of $\{\mathfrak{M}_{\mathfrak{X}} \Rightarrow Z\} \subset \{\mathfrak{M}_{\mathfrak{X} \cup \mathfrak{Y}} \Rightarrow Z\}$.

Next, we consider the other models $\mathfrak{C} := \{\mathfrak{M}_{\mathfrak{X} \cup \mathfrak{Y}} \Rightarrow Z\} \setminus M'$. First, there is the obvious restriction that $C \in \mathfrak{C}$ is not allowed to contradict Z. We already ruled this out in Eq. 10. Let us look at the following sample ranking functions:

$$\kappa_{gen} = \left\| \begin{array}{l} 212 \\ \hline 211 \\ 221 \\ 222 \end{array} \right\| \text{ and } \kappa_{\overline{gen}} = \left\| \begin{array}{l} 211 \\ \hline 222 \\ 212 \\ 221 \end{array} \right\| \tag{12}$$

We can easily see that κ_{gen} believes $2_a \Rightarrow 2_c$ since $\kappa_{gen}(2_a \Rightarrow 2_c) = 0$, but at the same time $\kappa_{gen}((2_a \wedge 2_b) \Rightarrow 2_c) = 1 > 0$. A revision with $2_a \Rightarrow 2_c$ using Eq. 7 would not change κ_{gen} at all.

The same issue occurs considering a revision with a conditional that has a negated consequent, such as $2_a \Rightarrow \overline{2_c}$. We show this for $\kappa_{\overline{gen}}$ which believes in this conditional and would not be changed by a revision with $2_a \Rightarrow \overline{2_c}$ using Eq. 7. Nevertheless it does not believe $(2_a \wedge 2_b) \Rightarrow \overline{2_c}$. We conclude that Eq. 7 does not produce a ranking function that is capable of generalization.

7 An Alternative Revision

Because of the described drawbacks we suggest an alternative revision technique. The proposed revision introduced in this section, $(\kappa \star (A|S))$, utilizes a new operator $\kappa[A]$ which returns the highest disbelief among all models of A. After a revision of κ with the conditional $S \Rightarrow A$, we still want the equivalent of Eq. 6 to hold:

$$(\kappa \star (A|S))(SA) < (\kappa \star (A|S))(S\overline{A}) \tag{13}$$

This is investigated in the following

Theorem 2 *If $\kappa(SA) \geq \kappa(S\overline{A})$, then the ranking function κ' derived from κ by rearranging the models using*

$$\begin{aligned} \forall M \in \mathfrak{M} : \kappa'(M) &:= (\kappa \star (A|S))(M) \\ &= \begin{cases} \kappa(M) - \kappa(S \Rightarrow A) & : M \models S \Rightarrow A \\ a' + b' & : M \models S\overline{A} \end{cases} \end{aligned} \tag{14}$$

with

$$\begin{aligned} a' &= \kappa[SA] - \kappa(S \Rightarrow A) + 1 \\ b' &= \kappa(M) - \kappa(S\overline{A}) \end{aligned}$$

will result in $\kappa'(SA) < \kappa'(S\overline{A})$. Consequently, κ' expresses the belief in $S \Rightarrow A$.

Proof Let $\kappa_1 := (\kappa * (A|S))$ and $\kappa_2 := (\kappa \star (A|S))$. Since $\kappa[A] \geq \kappa(A)$, by application of Theorem 1 can be deduced that $\kappa_2(SA) = \kappa_1(SA) < \kappa_1(S\overline{A}) \leq \kappa_2(S\overline{A})$. □

So, concerning the preservation of current belief, this method works just as good as Eq. 7, but introduces greater changes. In the following discussion of the properties of $(\kappa \star (A|S))$ with respect to negation and generalization we justify these changes. First, we consider negation.

Theorem 3 *Let $\mathtt{t} \in \mathfrak{D}_a$ and $\kappa' := (\kappa \star (\overline{\mathtt{t}_a}|S))$. Then*

$$\forall \mathtt{r} \in \mathfrak{D}_a \setminus \mathtt{t} : \kappa'(S \Rightarrow \mathtt{r}_a) \leq \kappa'(S \Rightarrow \mathtt{t}_a). \tag{15}$$

Proof By applying Eq. 14, we obtain $\kappa'(S\mathtt{t}_a) > \kappa'[S\overline{\mathtt{t}_a}]$. This is equivalent to

$$\forall \mathtt{r} \in \mathfrak{D}_a \setminus \mathtt{t} : \kappa'(S\mathtt{r}_a) < \kappa'(S\mathtt{t}_a).$$

Hence, if S is believed, the inequality of Eq. 15 will hold strictly. On the other hand, if $\overline{S}$ is believed, then $\kappa'(S \Rightarrow \mathtt{r}_a) = 0$ as well as $\kappa'(S \Rightarrow \mathtt{t}_a) = 0$. □

Theorem 3 induces that the observed inconsistency described in Sect. 5 does not appear after the repeated application of Eq. 14. Indeed, a revision of κ_{neg} with $(\overline{1_b}|1_a)$ now results in

$$\kappa'_{neg} = (\kappa_{neg} \star (\overline{1_b}|1_a)) = \left\| \begin{array}{c} 21 \\ \hline \\ 22 \\ 12 \\ 23 \\ 13 \\ 11 \end{array} \right\| . \tag{16}$$

Also,

$$\kappa''_{neg} = (\kappa'_{neg} \star (\overline{2_b}|1_a)) = \left\| \begin{array}{c} 21 \\ \hline \\ 22 \\ \\ 23 \\ 13 \\ 11 \\ 12 \end{array} \right\| \tag{17}$$

which illustrates that $1_a \Rightarrow 3_b$ is now believed as expected.

We now consider generalization with our alternative revision technique. Again, Eqs. 10 and 11 are given.

Theorem 4 *Let* $X \in \mathfrak{M}_X$, $Z \in \mathfrak{M}_Z$, *and* $\kappa' := (\kappa \star (Z|X))$. *Then*

$$\forall Y \in \mathfrak{M}_Y : \kappa'(X \wedge Y \Rightarrow Z) \leq \kappa'(X \wedge Y \Rightarrow \overline{Z}). \tag{18}$$

Proof The proof is an analog of the proof of Theorem 3. After applying Eq. 14, we obtain $\kappa'(X\overline{Z}) > \kappa'[XZ]$. This is equivalent to

$$\forall Y \in \mathfrak{M}_Y : \kappa'(X \wedge Y \wedge Z) < \kappa'(X \wedge Y \wedge \overline{Z})$$

Hence, if $X \wedge Y$ is believed, the inequality of Eq. 18 will hold strictly. On the other hand, if $\overline{X \wedge Y}$ is believed, then $\kappa'(X \wedge Y \Rightarrow Z) = 0$ as well as $\kappa'(X \wedge Y \Rightarrow \overline{Z}) = 0$. □

A similar theorem will hold, if the consequent is negated. To complete this section, we show that the previous counter-examples can be resolved using Eq. 14. A revision of κ_{gen} with $(2_c|2_a)$ now yields

$$\kappa_{gen} = \left\| \begin{array}{c} 212 \\ \hline 211 \\ 221 \\ 222 \\ \\ \\ \end{array} \right\| \xrightarrow{(\kappa_{gen} \star (2_c|2_a))} \kappa'_{gen} = \left\| \begin{array}{c} 212 \\ \hline \\ \\ 222 \\ 211 \\ 221 \end{array} \right\| . \tag{19}$$

This κ'_{gen} expresses a belief in $2_a \wedge 1_b \Rightarrow 2_c$ and $2_a \wedge 2_b \Rightarrow 2_c$.

A revision of $\kappa_{\overline{gen}}$ with $2_a \Rightarrow \overline{2_c}$ provides

$$\kappa_{\overline{gen}} = \left\| \begin{array}{c} 211 \\ \hline 222 \\ 212 \\ 221 \\ \\ \\ \end{array} \right\| \xrightarrow{(\kappa_{\overline{gen}} \star (\overline{2_c}|2_a))} \kappa'_{\overline{gen}} = \left\| \begin{array}{c} 211 \\ \hline \\ \\ 221 \\ 222 \\ 212 \end{array} \right\| \tag{20}$$

which now believes $2_a \wedge 1_b \Rightarrow \overline{2_c}$ as well as $2_a \wedge 2_b \Rightarrow \overline{2_c}$.

8 Application

We now examine the effect of the proposed algorithm in a gridword application. For this application, six cases are examined.

1. plain Q-learning
2. ranking-function-augmented Q-learning with application of Eq. 7
3. ranking-function-augmented Q-learning with application of Eq. 14
4. plain Q-learning with futile information
5. ranking-function-augmented Q-learning with application of Eq. 7 and futile information
6. ranking-function-augmented Q-learning with application of Eq. 14 and futile information

Ranking-function-augmented Q-learning means Q-learning where conditionals are extracted from the Q-Table. These conditionals revise the agent's ranking function and this ranking function acts as a filter for the actions afterward. This filter is implemented as a two-stage ϵ-greedy policy, in which the first stage decides whether or not the actions should be filtered and the second stage decides on the greediness. In both stages, we set $\epsilon = 0.1$. Figure 1 depicts this architecture.

We add futile information to model the case where the agent perceives properties of its environment that are not helpful with regard to its actual goal. This enlarges the state space and we therefore expect the plain Q-learner to perform worse in such an environment. The ranking-function-augmented Q-learners should be able to generalize and therefore identify the futile bits.

The generalization is performed in the same manner as in [13] by counting the pattern frequency. The general idea is to keep track of how often sub-patterns of antecedents are used in the context of particular consequents. If a sub-pattern occurs frequently enough, we revise the ranking function with that sub-pattern instead of the complete state description. This is

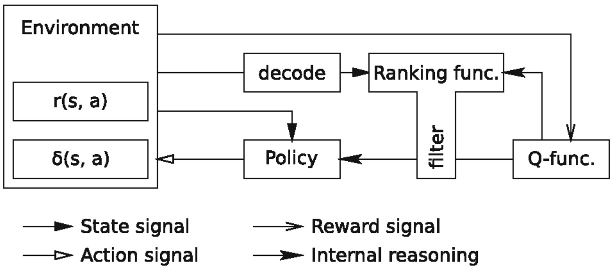

Fig. 1 Augmentation with a ranking function

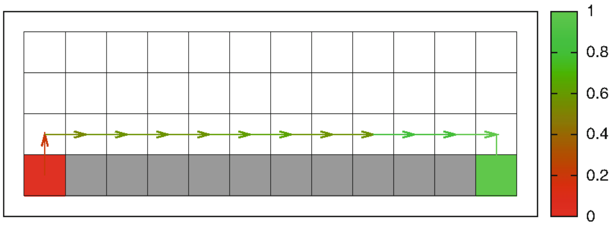

Fig. 2 The cliff-walk gridworld; superimposed the learned path after 100 episodes. The path color indicates the expected reward by displaying the value of $\min(1, \frac{\text{expected reward}}{\text{goal reward}})$ using the displayed color key

the point, where the behavior of the revision of ranking functions in the light of multi-valued logic and generalization matters.

The state description is also taken form [13] and consists of the following variables.

$$\begin{aligned}
\text{target lies north/south: } & t_{NS} \in \{\text{neither, N-ahead, N-aside, S-ahead, S-aside}\}\\
\text{target lies east/west: } & t_{EW} \in \{\text{neither, E-ahead, E-aside, W-ahead, W-aside}\}\\
\text{target distance: } & t_D \in \{\text{far } (\geq 5), \text{ middle } (< 5), \text{ close } (< 2)\}\\
\text{obstacle north: } & o_N \in \{\text{true, false}\}\\
\text{obstacle east: } & o_E \in \{\text{true, false}\}\\
\text{obstacle south: } & o_S \in \{\text{true, false}\}\\
\text{obstacle west: } & o_W \in \{\text{true, false}\}\\
\text{color: } & c \in \{\text{black, white, red, green, blue, yellow}\}\\
\text{action: } & a \in \{\text{go-west, go-east, go-north, go-south}\}
\end{aligned}$$

The gridworld itself is the cliff-walker example from [22]. There it has been used to illustrate the difference between on- and off-policy learning. Figure 2 depicts the set-up. The goal is to reach the green square, starting from the red one. Entering the black squares (chasm!) results in a large negative reward. The main parameters in a nutshell are:

$$\begin{array}{ll}
\text{reward for reaching goal} & = \ 100\\
\text{reward for a single step towards the goal} & = \ 0.5\\
\text{reward for every other single step} & = \ -1\\
\text{reward for stepping into the chasm} & = -10\\
\text{maximum number of steps in each episode} & = \ 100
\end{array} \tag{21}$$

Whether a step takes the agent towards the goal is measured by applying the Manhattan metric. Recording the reward over 300 episodes, averaged several times, yields the results depicted in Fig. 3.

It is evident that revising with Eq. 14 clearly surpasses revising with Eq. 7. The latter is worse than a plain Q-learner and even seems to degrade in performance over time. An explanation of this behavior may be that the ranking function gets contaminated by harmful conditionals. However, this has not been examined further in this work.

Let us risk a peak into the rules the agent has established using Eq. 14 and discuss their meaning. The most general rules learned are

$$\text{S-ahead}_{t_{NS}} \wedge \text{neither}_{t_{EW}} \Rightarrow \text{go-down}_a \tag{22}$$

$$\text{S-aside}_{t_{NS}} \wedge \text{E-ahead}_{t_{EW}} \Rightarrow \text{go-right}_a \tag{23}$$

$$\text{neither}_{t_{NS}} \wedge \text{E-ahead}_{t_{EW}} \Rightarrow \overline{\text{go-right}_a}. \tag{24}$$

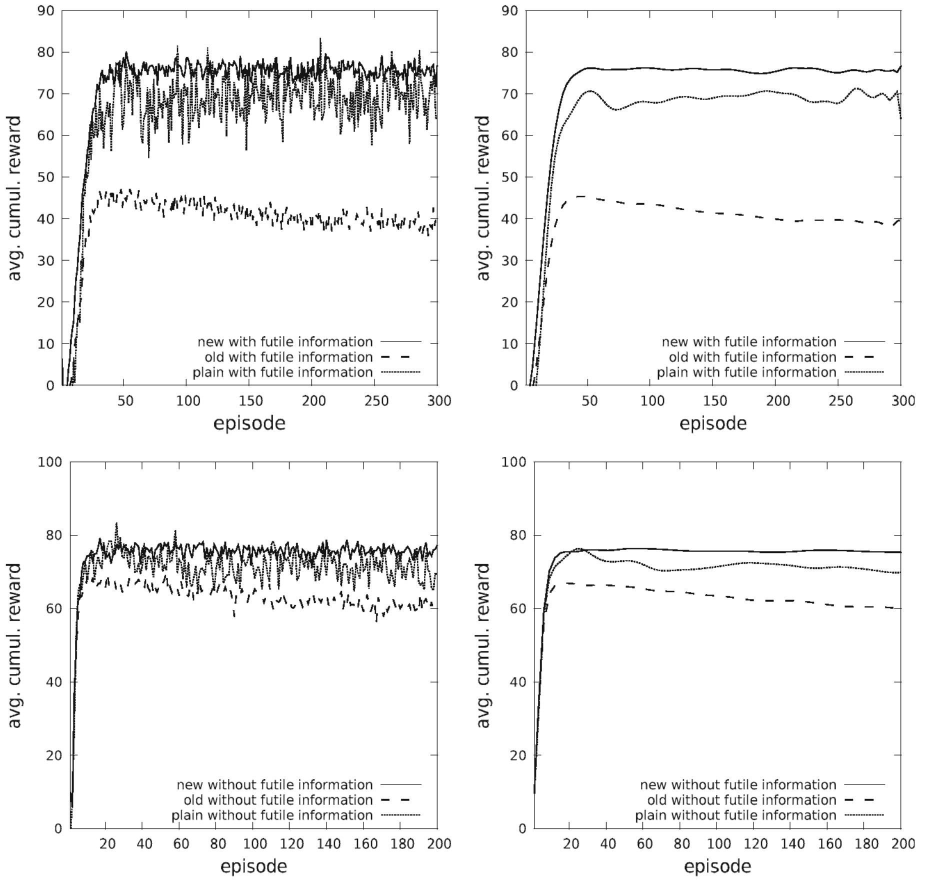

Fig. 3 The diagrams show the rewards over the course of 300 episodes. The left diagrams show the averaged results of 1000 runs for the ranking-function-augmented learners. Since the plain Q-learner exhibits a large variation, its curve has been averaged 2000 times. Because of the still strong variations, the right diagrams have been added that show a bezier-smoothed version of the left data. The *new*-curves are results of an application of Eq. 14 with and without futile information, the *old*-curves show results of Eq. 7 with and without futile information, and the *plain*-curves show results of a plain Q-learner with and without futile information

Equation 22 applies for the rightmost column of the gridworld and tells the agent to go down towards the goal. Equation 23 tells the agent to go right if the goal is in the south-east direction. This applies for the most part of the gridworld above the chasm row, except for the upper right triangle for which the description would be S-ahead$_{t_{NS}}$ $\wedge$ E-aside$_{t_{EW}}$. Finally, Eq. 24 applies at the starting position and prevents the agent from falling into the chasm right away.

Note the complete absence of obstacle descriptions. The relative position to the goal seems to be perfectly sufficient to determine the next best action.

The computational cost of the described improvement depends on the representation of the ranking function. If the ranking function is implemented by creating every possible conjunction beforehand, then Eqs. 7 and 14 will lead to roughly the same running time, because $\kappa[A]$ is the rank of A in a reversed κ. Unfortunately, the number of conjunctions has a combinatorial growth rate with regard to the number of variables in the state description. Our very small gridworld example already has 4800 possible conjunction without and 28800 conjunction with the futile information.

In a different approach we initialized the ranking function without any conjunctions to be able to handle larger problems. Conjunctions not occurring in the ranking function received a rank of infinity. Then the revision process generates conjunctions as needed. Clearly, this breaks the symmetry between $\kappa(A)$ and $\kappa[A]$. This approach is described in more detail in [11]. Code is carried out to generate conjunctions that comply with a given formula minus those that are already in the ranking function. When computing $\kappa[A]$ for some A we have to scan the whole ranking function and count the number of compliant conjunctions $|\{M|M \models A\}|$ to find out if there are any left with rank infinity. The runtime of this approach is about 1.5 times larger than the runtime of the previously described approach.

9 Conclusion

We presented a self-learning agent which exhibits all the basic properties we consider necessary for such a system [14]: hierarchical learning, emerging mechanisms, multi-directional transfer, generalization, exploration, and adaptivity.

Exploration, adaptivity, and emerging mechanisms are already properties of reinforcement learning agents. Since we build on reinforcement learning, our agent obviously exhibits them. The architecture we chose creates a hierarchical learning system, where the levels of the hierarchy communicate. The bottom-up communication is implemented via the creation of conditionals from the numerical values of the Q-table. Hence, the lower level generates the information which revises the higher level. The top-down direction, on the other hand, is only given implicitly through a filtering of available actions by the higher level rules. The presence of a higher symbolical level also allowed us to inspect and interpret the learned rules.

The main improvements this particular agent exhibits were obtained by analyzing the to-date revision operator of Spohn's ranking functions. By investigating a number of important cases which appear in a reinforcement learning agent concerning iterated belief revision, we were able to identify an improved revision method. Some apparent implausibilities concerning unobserved variables and negations were found to be resolved by our approach.

We did not discuss the application of different belief strength in this work. In general, one may specify the strength with which a given conditional may be believed by a ranking function as a second parameter of the revision operator. In this work, the belief strength is implicitly set to one. However, one can imagine an agent which may revise a conditional with a strength depending on its Q-value.

Because the agent is able to generate a symbolical belief representation from experience, it lays a foundation on which more elaborate methods of reasoning and inference may be applied. Since ranking functions are closely connected to Bayesian networks [18], recent advances [23] in the field may be transferred to such agents. However one may also proceed to apply long-studied techniques from the field of theorem proving [16].

In either case, how this can be used to further improve the agents performance needs to be examined in future research. Also, the application of this approach to both, more complex and more practical domains needs to be studied. We made a first step towards a more real-world application by applying the two-level-learning architecture to a task of object recognition, in which the agent had to distinguish between similar three-dimensional objects [10].

Acknowledgements This research was funded by the German Research Association (DFG) under Grant PE 887/3-3.

References

1. Alchourron CE, Gardenfors P, Makinson D (1985) On the logic of theory change partial meet contraction and revision functions. J Symbol Log 50(2):510–530
2. Anderson JR (1983) The architecture of cognition. Hardvard University Press, Cambridge
3. Blockeel H, De Raedt L (1998) Top-down induction of first-order logical decision trees. Artif Intell 101:285–297
4. Darwiche A, Pearl J (1996) On the logic of iterated belief revision. Artif Intell 89:1–29
5. Driessens K, Ramon J (2003) Relational instance based regression for relational reinforcement learning. In: Proceedings of the twentieth international conference on machine learning, pp 123–130
6. Dzeroski S, De Raedt L, Driessens K (2001) Relational reinforcement learning. Mach Learn 43:7–52
7. Gartner T, Driessens K, Ramon J (2003) Graph kernels and gaussian processes for relational reinforcement learning. In: Inductive logic programming, 13th international conference, ILP
8. Gombert JE (2003) Implicit and explicit learning to read: implication as for subtypes of dyslexia. Curr Psychol Lett 1(10)
9. Häming K, Peters G (2010) An alternative approach to the revision of ordinal conditional functions in the context of multi-valued logic. In: Diamantaras K, Duch W, Iliadis LS (eds) 20th international conference on artificial neural networks, September 15–18. Springer, Thessaloniki, pp 200–203
10. Häming K, Peters G (2011) A hybrid learning system for object recognition. In: 8th international conference on informatics in control, automation, and robotics (ICINCO 2011), Noordwijkerhout, The Netherlands, July 28–31
11. Häming K, Peters G (2011) Ranking functions in large state spaces. In: 7th international conference on artificial intelligence applications and innovations (AIAI 2011), September 15–18, Corfu, Greece
12. Kern-Isberner G (2001) Conditionals in nonmonotonic reasoning and belief revision: considering conditionals as agents. Springer, New York
13. Leopold T, Kern Isberner G, Peters G,(2008) Combining reinforcement learning and belief revision: a learning system for active vision. In: Everingham M, Needham C, Fraile R (eds) 19th British machine vision conference (BMVC 2008), September 1–4, vol 1. Leeds, UK, pp 473–482
14. Peters G (2011)Six necessary qualities of self-learning systems—a short brainstorming. In: International conference on neural computation theory and applications (NCTA 2011), October, Paris, France, pp 24–26
15. Reber AS (1989) Implicit learning and tacit knowledge. J Exper Psycol Gen 3(118):219–235
16. Robinson JA, Voronkov A (eds) (2001) Handbook of automated reasoning (in 2 volumes). Elsevier, New York
17. Spohn W (August 1988) Ordinal conditional functions: a dynamic theory of epistemic states. In: Causation in decision, belief change and statistics, pp 105–134
18. Spohn W (2009) A survey of ranking theory. In: Degrees of belief. Springer, New York
19. Sun R, Merrill E, Peterson T (2001) From implicit skills to explicit knowledge: a bottom-up model of skill learning. Cogn Sci 25:203–244
20. Sun R, Terry C, Slusarz P (2005) The interaction of the explicit and the implicit in skill learning a dual-process approach. Psychol Rev 112:159–192
21. Sun R, Zhang X, Slusarz P, Mathews R (2006) The interaction of implicit learning, explicit hypothesis testing, and implicit-to-explicit knowledge extraction. Neural Netw 1:34–47
22. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge
23. Tenenbaum JB, Kemp C, Griffiths TL, Goodman ND (2011) How to grow a mind: statistics, structure, and abstraction. Science 331(6022):1279–1285
24. Ye C, Yung NHC, Wang D (2003) A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance. IEEE Trans Syst Man Cybern B 33(1):17–27