Iterated Revisions in Large and Noisy State Spaces Using Ranking Functions

Klaus Häming and Gabriele Peters

University of Hagen,
Chair of Human-Computer-Interaction,
Universitätsstr. 1, 58097 Hagen, Germany
{klaus.haeming,gabriele.peters}@fernuni-hagen.de

Abstract

Ranking functions rank all possible instances of a world according to the disbelief associated with each instance. Two problems follow from this simple premise. The first is that all instances of a world can only be enumerated for very small worlds. The second issue arises whenever ranking functions are used to represent the belief of an actual agent. Since the observations of that agent are usually contaminated with noise, it is impossible to directly relate them to a particular world instance. This work discusses how to cope with these difficulties. Additionally, the proposed methods are assessed in a reinforcement learning application, where the usage of a ranking functions enables an agent to learn where it would fail otherwise.

Keywords: ranking functions, reinforcement learning, belief revision, computer vision

1 Introduction

This work is about improving the capabilities of an autonomous agent to learn from experience. The agent under consideration lives in an environment whose states it perceives. For each state, it has to choose from a number of possible actions. The effect of those actions are initially unknown to the agent, but yield a reward once chosen.

A problem of this kind is usually approached by applying reinforcement learning [23]. In reinforcement learning, the agent incorporates its experience and observations into its belief state in order to learn and solve its task successfully. The experience usually consists of rewards which follow from certain actions. Traditionally, reinforcement learning approaches work by building up a belief about the workings of the environment. This belief is represented, for instance, by using a Q-function, which numerically represents the expected benefit of the available actions in a given state.

This approach leads to the question of how to represent this Q-function. Naturally, the first constraint arises in real-world applications whose state spaces are usually described by more than a few variables. Since a state space grows exponentially with the number of variables, a representation of the Q-function is needed which can capture the experiences in a compact manner. For this purpose any kind of supervised learning [4] may be applied.

A second problem arises, when states are described symbolically. In this case, the symbolic representation has to be mapped to a numerical one in order to match the requirements of the traditional reinforcement learning framework. The mapped values are henceforth combined into new values, thereby loosing their symbolical meaning (depending on the mapping). Efforts to make reinforcement learning able to handle symbolic representations can be found, e.g., in the context of relational reinforcement learning [7]. Approaches include a representation as decision trees [3], defining a distance function on predicates to allow nearest neighbor interpolation [6], and applying a kernel approach [8].

A different approach combines two different representations. One representation thereof is numerical, the other symbolical. This approach is inspired by psychological findings [9, 16]. These findings suggest that there are essentially two kinds of learning, a low-level implicit learning and a high-level explicit learning. In this area, the work of Sun [21, 22], who encoded the symbolical representation in a neural network, is an example.

We also follow the two-level-learning path. While the lower, still numerical level is represented by a Q-function, we propose to use Spohn's ranking functions as a natural way to represent the symbolical level. Spohn's ranking functions were introduced under the term of ordinal conditional functions [17]. They were introduced to account for the dynamics of belief revision [1, 5]. Traditionally, belief revision deals with belief sets, which capture the current belief of an agent. A belief set changes whenever an agent perceives new information it wants to include. A well accepted set of postulates which constrain the possible change of a belief set is given by the AGM theory of Alchourrón, Gärdenfors and Makinson [1]. Ranking functions are compliant with these postulates [18], but they extend the belief set in a way that, additional to the actual belief, the disbelieved facts are also included. This property allows ranking functions to account effectively for changes in belief and also allows for an elegant incorporation of conditional information. As Spohn discusses in [20], ranking functions are also closely related to Bayesian inference [2].

Spohn's ranking functions are not to be confused with ranking functions that aid a search algorithms in a heuristic manner. An example for the latter kind of ranking function is the well-known tf–idf weight [24] which is often used in text mining. These ranking functions are a tool to *cope* with large state spaces, as shown in, e.g., [25]. The ranking function we discuss in this work rank world models and hence will suffer from large state spaces *themselves*, if we do not take steps against it. In this work, we investigate the applicability of ranking functions on problem domains which indeed have a large state space. Additionally, we propose an approach for an agent who lives in such a domain and whose observations are noisy or incomplete. Because the belief state is represented as a ranking function, a large part of this work discusses ranking functions and their problems with large state spaces independently from a concrete application.

The next section provides a brief review of Spohn's ranking functions and the mechanics of a revision with propositional information. The revision with conditional information is discussed thereafter, followed by a discussion of the problems with large state spaces. Then, after we present our proposal of how to handle these problems, we proceed to a discussion of the problems noise and uncertainty pose for a belief representation. This part is heavily biased towards our example application, which is introduced thereafter. After that, a brief section with concluding remarks is given.

2 Ranking Functions

To define ranking functions, we need the notion of a *world model*. Given the set of all possible worlds $\mathfrak{W}$, a world model $M \in \mathfrak{W}$ describes exactly one instance. Therefore, to apply ranking functions, the world an agent lives in must be completely describable. Initially, we assume such a world.

A particular ranking function $\kappa : \mathfrak{W} \to \mathbb{N}$ assigns a non-negative integer value to each of the world models. These *ranks* reflect the amount of *disbelief* an agent shows for each model. It is perfectly possible that rank 0 is assigned to both, a world model *and* its negation. Therefore we say that the agent believes in a model M, iff

$$\kappa(M) = 0 \wedge \kappa(\overline{M}) > 0\ . \qquad (1)$$

Besides querying the rank of a world model, we may ask for the rank of a more general *formula*. For instance, in a world described by two variables a and b, each from the domain $\{1, 2, 3\}$, a formula F may look like

$$F = ((a = 2) \vee (a = 3)) \wedge (b = 1). \qquad (2)$$

Since a model M captures all aspects of the world the agent lives in, F partitions $\mathfrak{W}$ into two sets:

1. $\{M | M \models F\}$
2. $\{M | M \models \overline{F}\}$

This allows us to define the rank $\kappa(F)$ of a formula F as

$$\kappa(F) = \min\{\kappa(M) | M \models F\}. \tag{3}$$

So, if an agent believes in a particular world model which happens to entail F, it will also believe F.

For example, consider Equation 1 again. There, $\overline{M}$, the negation of world model M is used. This negation is a formula. Since a particular world model M is a conjunction of an assignment of all available variables, $\overline{M}$ is entailed by all models that disagree with M on at least one variable assignment. Obviously, these are *all* other models and hence we can conclude that the agent believes in a particular world model M iff M is the only one which is mapped to rank 0.

Whenever the agent experiences new information about its world, new ranks are assigned to the world models to reflect the agent's new belief state. This incorporation of new information is called *revision*. In the context of revision, a peculiar property of ranking functions is their ability to not only belief in a certain proposition, but to belief in it with a given strength $\beta \in \mathbb{N}$, which is enforced during revision. The strength $\beta(F)$ of the belief in a formula F can be calculated as

$$\beta(F) = \kappa(\overline{F}) - \kappa(F). \tag{4}$$

Note that by using this formula it is possible—and probably more intuitive—to reformulate ranking function in terms of belief instead of disbelief [19]. In this discussion, however, we prefer a formulation in terms of disbelief.

We discuss two types of revision. First, the revision with propositional information, and second the revision with conditional information. After each definition, a short explanation of its meaning is given.

Definition 1 *Given a proposition P and a strength parameter β, the revision $\kappa * (P,\beta)$ is given by*

$$\kappa * (P,\beta)(M) = \begin{cases} \kappa(M) & : \kappa(\overline{P}) \geq \beta \\ \kappa(M) - \kappa(P) & : \kappa(\overline{P}) < \beta \wedge M \models P \\ \kappa(M) + \beta - \kappa(\overline{P}) & : \kappa(\overline{P}) < \beta \wedge M \models \overline{P} \end{cases} \tag{5}$$

Essentially this definition states that

1. there is nothing to do, if P is already believed with strength β
2. otherwise two subsets of the set of world models are relevant:
   (a) the subset of those world models which agree on P, and whose ranks are modified so that we assign rank 0 to the least disbelieved model.

(b) the subset of those world models which agree on $\overline{P}$, and whose ranks are modified so that $\kappa(\overline{P}) \geq \beta$ holds afterwards.

The rationale behind this partitioning is that the world models within $\{M|M \models P\}$ are independent *conditional* on P and should therefore keep their *relative* ranks. Because the only non-constant value in Equation 5 is the current rank $\kappa(M)$ of a world model M, it is obvious that the relative ranks within the subsets are preserved. The same argument is applicable to $\{M|M \models \overline{P}\}$ and $\overline{P}$.

Because we know that within these two subsets the relative ranks are preserved, considering the lowest ranked world model suffices to understand Definition 1:

1. the rank of the lowest ranked world model M_P of $\{M|M \models P\}$ is just $\kappa(P)$, so that $\kappa * (P,\beta)(M_P) = \kappa(M_P) - \kappa(P) = 0$
2. the rank of the lowest ranked world model $M_{\overline{P}}$ of $\{M|M \models \overline{P}\}$ is just $\kappa(\overline{P})$, so that $\kappa * (P,\beta)(M_{\overline{P}}) = \kappa(M_{\overline{P}}) + \beta - \kappa(\overline{P}) = \beta$

In the following example, we use a table to depict a ranking function. In such a table, the top row contains the models mapped to rank 0. Higher ranks follow in order. The models are included by writing the values of the variables down. The order of the values matches a given order of the variables, which is usually lexicographical and should otherwise be obvious from the respective context.

The table of Equation 6 shows the top three ranks before and after a revision takes place. The ranking function contains world models of a world consisting of two variables. These variables and their domains are $a \in \{1, 2, 3\}$ and $b \in \{1, 2\}$. After the revision, the ranking function on the left shall believe in $(a = 3)$ with strength 1:

κ	$\xrightarrow{\kappa*(a=3,1)}$	κ'
21		32
12 22 32	$\rightarrow$	21 31
31		12 22
$\vdots$		$\vdots$

(6)

It can be seen that the two world models which comply with $(a = 3)$ are $(a = 3) \wedge (b = 2)$ and $(a = 3) \wedge (b = 1)$. These are shifted such that $(a = 3)$ is believed in terms of Equation 1. The group of world models that does not comply with $(a = 3)$ consists of $(a = 2) \wedge (b = 1)$, $(a = 1) \wedge (b = 2)$, and $(a = 2) \wedge (b = 2)$. This group is shifted one rank down. Note that at the same time the relative ranks are preserved within the two groups of world models.

3 Revising Conditionals

After the gentle introduction to revising with a proposition, we now focus on revising with a conditional $(B|A)$ with A the antecedent and B the consequent. After such a revision we expect an agent to show the following property: if it is subsequently revised with A then it must also believe in B, because this is what the conditional states. We may write this as

$$\kappa'(B) = 0 \wedge \kappa'(\overline{B}) > 0 \text{ with } \kappa' = (\kappa * (B|A)) * A. \tag{7}$$

At first sight, we may just apply Definition 1 using the formula of material implication, $A \Rightarrow B$, because this is the formula which represents a conditional dependency. The following example ranking function shows that this is not sufficient:

$$\begin{array}{c} \kappa \\ \hline \left\| \begin{array}{c} 2\,2 \\ \hline 1\,2 \\ 1\,3 \\ \vdots \end{array} \right\| \end{array} \tag{8}$$

Revising κ of Equation 8 with the formula $F = (a = 1) \Rightarrow (b = 3)$ will not change anything since F is already believed, because $(a = 2) \wedge (b = 2) \models \overline{(a = 1)} \vee (b = 3)$. But Equation 7 does not hold for this ranking function. Hence, we clearly need a different approach.

In [14] a revision was proposed which does not share the weakness of the aforementioned simple approach. But is has problems with iterated revisions, which was shown in [12]. There we proposed to use the additional operator

$$\kappa[F] := \max\{\kappa(M) | M \models F\} \tag{9}$$

to define the revision with a conditional as follows:

Definition 2 *Given a conditional* $(B|A)$*, the revision* $\kappa * (B|A, \beta)$ *is given by*

$$\kappa * (B|A,\beta)(M) =$$
$$\begin{cases} \kappa(M) & : D \geq \beta \\ \kappa(M) - \kappa(A \Rightarrow B) & : D < \beta \wedge M \models (A \Rightarrow B) \\ \kappa(M) + \underbrace{(\kappa[AB] - \kappa(A \Rightarrow B) + \beta)}_{\beta'} - \kappa(A\overline{B}) & : D < \beta \wedge M \models (A\overline{B}) \end{cases} \tag{10}$$

with $D = \kappa(A\overline{B}) - \kappa[AB]$ *.*

The general idea behind Definition 2 is the same as the one behind Definition 1: Partition the models into two groups and shift the ranks up and down until the required condition is reached!

The two definitions show even more striking similarities if we interpret the part above the horizontal brace as a modified strength parameter β'. To see what this β' does, imagine that M is the lowest ranked world model which entails $\kappa(A\overline{B})$. Also, let M_{AB} be the world model responsible for the value of $\kappa[AB]$ *before* the revision. After the revision, the rank of M_{AB} will be $\delta = \kappa[AB] - \kappa(A \Rightarrow B)$.

Let κ' denote the ranking function after the revision. Then, $\beta' = \beta + \delta$ means that $\kappa'(A\overline{B}) = \kappa'[AB] + \beta$, because then we have $\kappa'(A \Rightarrow B) = 0$. Hence, we may understand the application of Definition 2 by picturing the rank of $A\overline{B}$ as β below the rank of AB after the revision has affected κ.

The following example clarifies the belief change in the presence of conditional information. Suppose our world is described by three variables $a, b, c \in \{1, 2\}$. Suppose further that our current belief state represents a belief in $M_0 = (a = 2) \wedge (b = 1) \wedge (c = 1)$ which is consequently mapped to rank 0. The left column of the following table shows this scenario. Again, the first row contains world models with rank 0, i.e., the believed ones, which is in this example just M_0. There are a few other models whose respective ranks are 1, 2, and 3. The models are represented by the values of the variables in the order a, b, c. The models with higher ranks are omitted.

$$\begin{array}{c c c}
\kappa & \xrightarrow{\kappa * ((a=2)\Rightarrow(c=2),1)} & \kappa' \\
\hline
\begin{array}{||c||}
2\,1\,1 \\ \hline
2\,1\,2 \\
2\,2\,1 \\
2\,2\,2 \\
\\
\\
\vdots
\end{array}
& \rightarrow &
\begin{array}{||c||}
2\,1\,2 \\ \hline
\\
2\,2\,2 \\
2\,1\,1 \\
\\
2\,2\,1 \\
\vdots
\end{array}
\end{array} \tag{11}$$

The right column of this example captures the belief after κ has been revised with $(a = 2) \Rightarrow (c = 2)$ using Definition 2 (with strength $\beta = 1$). As you can see, the models have been partitioned into two groups ($\{2\,1\,2, 2\,2\,2\}$ and $\{2\,1\,1, 2\,2\,1\}$), which have been shifted such that κ obeys the postcondition $\kappa'(A\overline{B}) - \kappa'[AB] = 1$.

4 Large State Spaces

After the general mechanics of ranking functions have been described, we want to focus on the applicability of them on problems with a considerably large state space. Assume, for example, that a world is described by n boolean variables which describe binary properties of that world. A concatenation of these variables yields an n-digit binary number. The number of possible world models is therefore 2^n, which grows dramatically as n grows.

Therefore, for a sufficiently complex world, we cannot expect to be able to store all its possible models to represent κ:

κ			
M_1	M_2	M_3	$\cdots\lightning$
M_4	M_5		
$\vdots$			
$\lightning$			

We may try to circumvent this problem by modeling a skeptic agent, one that initially disbelieves everything. We may also think that this means we commit to a scenario, in which initially all models have rank ∞ and simply omit those from our representation. That would mean our initial ranking function is empty and the problem changes into being careful in putting new world models into it.

But there is a difficulty. As Spohn has shown, for each model M it must hold that

$$\kappa(M) = 0 \vee \kappa(\overline{M}) = 0 \ . \tag{12}$$

If we are to put every single model initially to rank ∞, this rule will be violated. Fortunately, the actual rank is less important than the fact whether a model is believed or not. Re-consider Equation 1. If neither a model M nor $\overline{M}$ has been experienced, we are also free to interpret this as

$$\kappa(M) = 0 \wedge \kappa(\overline{M}) = 0 \ , \tag{13}$$

which means that neither M nor $\overline{M}$ is believed. And once κ has been revised with M, the missing model $\overline{M}$ can be interpreted as $\kappa(\overline{M}) = \infty$, because that will always fulfill the postcondition of a belief strength larger than any given β. Hence, a skeptic agent indeed allows us to omit not yet experienced world models.

Concerning revision in large state spaces, we also need to investigate whether the *dynamics* induced by Definition 1 and Definition 2 create additional problems. First, revising with propositions is not problematic, as long as there are not too many models which entail it. Revising with propositional information is also not the common case for a learning agent, since the information is usually in conditional form as, e.g., in our example application below.

A Revision with conditional information, however, creates more serious problems. Let us assume that an agent needs to believe the formula $A \Rightarrow B$, therefore the models $\{M|M \models \overline{A} \vee B\}$ have to be shifted towards rank 0. If we re-consider the example of a world described by binary variables, revising with a conditional which is currently not believed and whose antecedent consists of just one variable means that at least half of the models have to be shifted towards rank 0. Hence, a lot of models not yet present in the (skeptic) ranking function have to be created in order to maintain their relative ranks. Furthermore, the operator $\kappa[AB]$ creates additional complications. If $\kappa[AB] = \infty$, we will also have to create all $\{M|M \models AB \wedge \kappa(M) = \infty\}$ regardless whether or not any models of AB are already known to the agent.

5 Coping with Large State Spaces

If we want to use ranking functions in large state spaces, we will obviously need a way around these issues—and we do have an option. The problem is caused by strictly requiring that within each of the subsets of $\mathfrak{W}$ (which are induced by a revision) the relative ranks stay the same. If we are allowed to bend this rule in order to instantiate the least amount of models, then we can proceed using ranking functions without hitting any serious computational barrier.

For a revision with propositional information P, the postcondition $\kappa(\overline{P}) \geq \beta$ must hold afterwards. We can comply with this requirement by just taking those models of P into account that have been experienced so far. This is true, because after shifting them towards rank 0, there either are models of $\overline{P}$ in the ranking function or not. If there are such models, the shifting of their ranks creates a ranking function as desired. If there are no such models, there is nothing to do because the postcondition already holds.

Similarly, for a revision with conditional information, we want the postcondition $\kappa(A\overline{B}) - \kappa[AB] \geq \beta$ to hold afterwards. So, if κ is to be revised with the formula $A \Rightarrow B$ and $\kappa(\overline{A} \vee B) = \infty$, then we may shift $\{M|M \models AB\}$ towards rank 0, regardless of other models in $\{M|M \models \overline{A} \vee B\}$. We also need to adjust the ranks of $\{M|M \models A\overline{B}\}$, of course, but these retain their ranks or are shifted to higher ranks, therefore posing no additional difficulties. We also modify $\kappa[AB]$ to take the form

$$\kappa[AB] = \max\{\kappa(M)|M \models AB \wedge \kappa(M) < \infty\}, \tag{14}$$

i.e., it returns the highest ranked *known* model of AB. Fortunately, this still leads to revisions after which the formula $A \Rightarrow B$ is believed, because this simply requires that *some* model of $\overline{A} \vee B$ has rank 0. Of course, there still is the case $\kappa(AB) = \infty \wedge \kappa(\overline{A} \vee B) \leq \infty$. In this case $\{M|M \models AB\}$ is shifted towards the highest known rank incremented by one. A drawback of this approach of considering only experienced world models, especially the modification of $\kappa[M]$, is that the iterated application of Definition 2 may lead to intuitively implausible results. These results are discussed in [12], where the choice of $\kappa[M]$ rather than $\kappa(M)$ is discussed.

An overview of the main properties of the proposed implementation concerning conditionals in ranking functions is given in Table 1. The important cases and their effect on a revision are listed. The left column contains properties of κ before and the right column the important changes after a revision with $(B|A)$. There "$\kappa(M) < \infty$" means that $\kappa(M)$ is finite.

κ	$\kappa * (B\|A)$
κ is empty	$\{M\|M \models AB\}$ are included $\kappa(AB) = \kappa[AB] = 0$ $\kappa(A\overline{B}) = \infty$
$\kappa(A \Rightarrow B) \leq \kappa[AB] < \kappa(A\overline{B}) = \infty$	no changes
$\kappa(A \Rightarrow B) \leq \kappa[AB] \leq \kappa(A\overline{B}) < \infty$	$\kappa(A\overline{B})$ may be increased depending on β
$\kappa(A \Rightarrow B) < \kappa[AB] = \kappa(A\overline{B}) = \infty$	$\{M\|M \models AB\}$ are included $\kappa(AB) = \kappa[AB] = \max_M(\kappa(M)) + 1$
$\kappa(A\overline{B}) < \kappa[AB] = \kappa(A \Rightarrow B) = \infty$	$\{M\|M \models AB\}$ are included $\kappa(AB) = \kappa[AB] = 0$ $\kappa(A\overline{B})$ may be increased depending on β
$\kappa(A\overline{B}) < \kappa(A \Rightarrow B) \leq \kappa[AB] < \infty$	$\kappa(A \Rightarrow B) = 0$ $\kappa(A\overline{B})$ will be increased
$\kappa(A\overline{B}) < \kappa(A \Rightarrow B) < \kappa[AB] = \infty$	$\{M\|M \models AB\}$ are included $\kappa(AB) = \kappa[AB] = \max_M(\kappa(M)) + 1$ $\kappa(A\overline{B})$ will be increased

Table 1: Effects of the proposed implementation.

6 Incomplete or Noisy Observations

Describing a world in terms of logical propositions leads to problems once the world starts to exhibit uncertainty. This section discusses a particular scenario in which this is the case.

Assume that an agent's state description consists of visual information. In particular, assume the state description enumerates a number of visual features $f_1, f_2, \ldots, f_n$. This scenario may happen in a computer vision application which uses a feature detector on images.

The problem is that feature detection is not reliable. The image is a quantized and noisy representation of the continuous world's electromagnetic signals. Viewing a scene twice from *nearly* identical positions may therefore yield slightly different descriptions of the visual input.

Assume the state description takes the form

$$S = v_1 \wedge v_2 \wedge v_3 \wedge v_4 \wedge \ldots v_n \ , \tag{15}$$

where v_i is either `TRUE` or `FALSE` depending on the visibility of the feature f_i. If the value of one of these v_i switches, the resulting state description will be completely different in terms of ranking functions. There is no definition of similarity.

Another problem is that it may be impossible to enumerate all models. Not because the state space is so large, but because it is practically the case that not all features are known beforehand but added as they appear:

$$\begin{array}{c} \kappa \\ \hline \left\| \begin{array}{c} M_1 = v_1 \wedge \overline{v_2} \wedge v_3 \wedge v_4 \wedge \overline{v_5} \wedge \cdots ? \cdots \lightning \\ \hline \vdots \end{array} \right\| \end{array}$$

As a consequence, we cannot create the partitions of $\mathfrak{W}$ necessary for revision.

In this particular example of a features-perceiving agent, it is possible to use ranking functions in spite of these issues. Both difficulties, the absence of a similarity definition and the unknown set of features is addressed by a modified entailment operator.

Before we introduce this operator, we have to make sure that for a revision with a conditional $(B|A)$ using the formula $A \Rightarrow B$ we are able to partition the models of $\mathfrak{W}$ into the two subsets $\{M|M \models \overline{A} \vee B\}$ and $\{M|M \models A\overline{B}\}$. This is required to apply Definition 2. To enforce this, we restrict the conditionals $(B|A)$ such that their associated formula $A \Rightarrow B$ is an element of the following set $\mathfrak{C}$:

Definition 3 $\mathfrak{C} := \{A \Rightarrow B | A \in \mathfrak{P} \neq \emptyset \wedge B \in \mathfrak{Q} \neq \emptyset \wedge \mathfrak{P} \cap \mathfrak{Q} = \emptyset\}$ *with* $\mathfrak{P}$ *and* $\mathfrak{Q}$ *chosen such that* $\forall M \in \mathfrak{W} : \exists P \in \mathfrak{P}, Q \in \mathfrak{Q} : M = P \wedge Q.$

So, the antecedents and consequents of the allowed conditionals are taken from different sets of variables which together contain all the world's variables. This allows us to handle the entailment differently for the antecedent and the consequent.

Back to our feature-based world, let us assume our models take the form $M = P \wedge Q$. P shall contain the unreliable, feature-based information, while Q covers everything else. Let us further assume that a function $F(P)$ exists which maps the feature-part P to all the features visible in M. This can be used to relate the visible features $F(P)$ and $F(P')$ of two models M and $M' = P' \wedge Q'$ to define an entailment operator $M \models_t M'$ such as in

Definition 4 $M \models_t M' :\Leftrightarrow \left(\frac{|F(P) \cap F(P')|}{|F(P) \cup F(P')|} > t\right) \wedge (Q \wedge Q')$

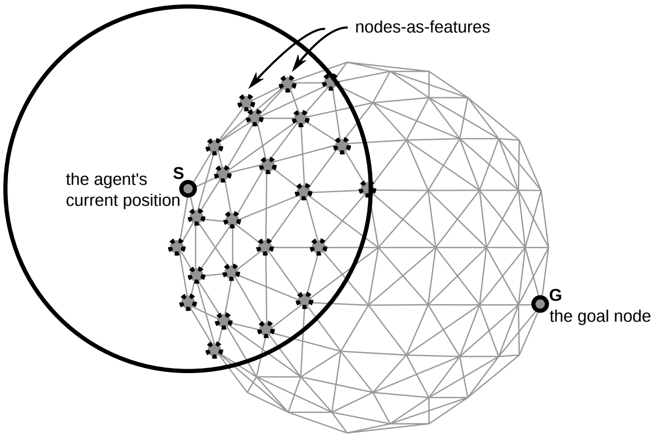

Figure 1: Simulated Features.

The left term thus defines a similarity on sets of visible features using the Jaccard index [13]. But how should t be chosen? This is assessed in the next section.

7 Example Application

To assess the applicability of the aforementioned ideas, we use an experimental set-up in which we present an environment to the agent that is perceived as a set of (unreliable) features as introduced in Section 6.

In this environment, the agent needs to find a spot on a spherical grid. The agent's state consists of a subset of all nodes present in a spherical neighborhood around its current location S, including S itself. Each of these nodes is classified as "visible" with a preset probability. The purpose of this experiment is to mimic the situation in which an object is perceived by a camera and represented by visual features. The visibility probability models feature misses and occlusions. Another way to see this experiment is to assume that the spherical grid itself is observed and the nodes are detected by an unreliable "junction detector".

In either case, the agent uses the information in its state signal about visible features to determine its position "above" the object. Figure 1 shows a picture of this set-up. We used a grid with 128 nodes placed on the surface of a unit sphere. The radius of the sphere that defines the neighborhood has been set to 1.2, which results in approximately 50 enclosed "features" at each position. An action is represented by a number which is associated with a grid node. One grid node is marked as being the goal node. Once the agent has reached this node, an episode ends.

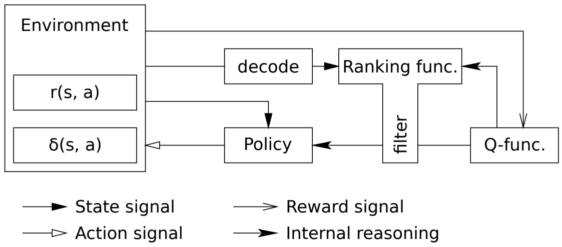

Figure 2: Two-level reinforcement learning set-up.

We used the two-level-learning approach of [10] and [15], in which a ranking function acts as a filter on the possible actions a reinforcement learning [23] agent may take. The reinforcement learning approach we used is Q-learning, which has the appealing property that it learns the optimal state-action-pairs regardless of the policy used for exploration. Figure 2 clarifies the architecture. There, δ is a transition function and r a reward function. The agent explores its environment, i.e. the nodes of the spherical grid, through trial and error until the goal state is reached. The goal state is defined as reaching the goal node, for which we choose the one farthest away from the starting location.

The arrival at the goal state triggers a reward of 100. All other actions are neither rewarded nor punished. In spite of this fact, the agent develops a preference for the shortest paths. This is due to our use of the discount factor $\gamma \in]0;1[$ which is present in the update equation

$$Q_{t+1}(S_t, A_t) = r(S_t, A_t) + \gamma \max_a (Q_t(\delta(S_t, A_t), A)). \tag{16}$$

This equation updates the Q-function of the Q-learning agent. There, S_t is the state at time t and A_t is the action chosen at that state. Because we only reward the final step to the goal state, the effect of γ is that states closer to the goal state receive a larger Q-value, thereby guiding the agent. We chose $\gamma = 0.6$.

The exploration policy is a modified ϵ-greedy policy with $\epsilon = 0.1$. The modification makes the policy aware of the filtering and gives it the option to ignore it once in a while. One may picture the policy as a cascaded ϵ-greedy policy, where the first step decides whether or not to accept the filtering and the second step decides on the greediness.

Figure 2 also shows the presence of a “decode” module. This module creates the symbolical state description from the state signal the environment provides. In our case this means essentially the enumeration of the visible features. The Q-function is realized by applying a hash function on the state description to create a suitable integer number. We want to stress the point that if a single number representing the node was used as the state description for the Q-table the result would obviously be more efficient. However, this would also remove the large state space and the noise in its description and would therefore render the experiment inappropriate to model visual input.

The conditionals used for revision are created with the aid of the Q-function. In each state S a revision with the conditional $(A|S)$ takes place, whenever a single best action A has been identified by the Q-function and the ranking function does not already believe in the associated conditional.

Figures 4, 5, and 6 show the averaged cumulated rewards and the averaged episode lengths for a number of experiments. Each figure captures the results for a particular value of the feature detection probability. The graphs show 200 episodes averaged over 500 runs. Each episode had to end after 100 steps or when the goal had been found. Depicted are the learning progresses for agents with different t-values as well as for a "plain" Q-learner. The latter declines the usage of a ranking function and hence the usage of a second learning level.

Figure 4 shows the results of setting the feature detection probability to 0.5. The best learning agent is the one with $t = 0$. Note that the large state space clearly hinders the plain Q-learner to show any progress within the 200 episodes. Confirming these observations, Figure 5 shows the results of setting the feature detection probability to 0.9. Despite the relatively high feature detection probability, the plain Q-learner is still not able to succeed.

To validate that the plain Q-learner is in principle able to eventually learn a short path to the goal state, we include Figure 6. In this experiment, all features are always observed. Hence, every state now has a unique description. In this scenario, a plain Q-learner shows improvements, but is still slower in reaching a high level than an agent augmented with a ranking function.

So, the answer to the question asked in Section 6 about a good value for t in Equation 4 is $t = 0$. This can be interpreted in such a way that the agent is able to identify its current state by recognizing one feature alone. This allows us to simplify Definition 4 to

Definition 5 $M \models_t M' :\Leftrightarrow (|F(P) \cap F(P')| > 0) \wedge (Q \wedge Q')$

Requiring it to recognize more features hinders the agent's learning and therefore slows down its progress.

A trained agent can use its experience to find its way towards the goal state. Some of the possible results obtained in our experiments are shown for each discussed feature detection probability in Figure 3. The paths were generated by asking the agent's ranking function for directions and choosing a random direction whenever the agent had no preference. All of theses paths lead more or less directly from start to goal. The shortest path was learned for the feature detection probability of 1.0, as one may expect. As in the other experiments, it took 200 episodes to train the agent.

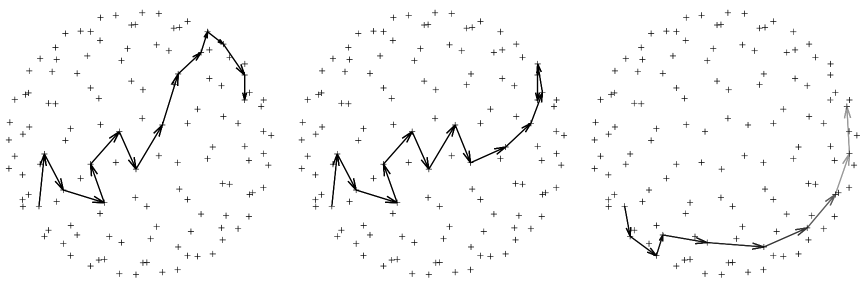

Figure 3: Learned paths for feature detection probabilities of 0.5, 0.9, and 1.0 (from left to right).

8 Discussion and Conclusion

This work discusses the applicability of ranking functions in two very important cases. These two cases are

1. large state spaces
2. noisy or incomplete observations.

Judging by the requirement of ranking functions to enumerate all possible world models, their usage seems highly questionable at first sight. However, a careful modification of the revision process to avoid an instantiation of a large number of models as well as the introduction of a domain specific entailment operator led to a system that was able to learn in a rather complex environment. In the same environment, an agent without a supplementing ranking function had failed.

Guided by our intend to use ranking functions in a reinforcement learning context, emphasis was laid upon the revision with conditional information. This kind of information occurs naturally in a context where observed states lead to actions. Therefore, our main concern was the revision $\kappa * (B|A)$. This includes also $\kappa * (\overline{B}|A)$ and $\kappa * (B|\overline{A})$, because changing the antecedent or consequent simply means a different set of agreeing world models. Also, in the case where we want an agent *not* to believe in a given conditional, we may just revise it with the proposition $A\overline{B}$.

Aside from philosophical discussions about belief revision and causation, ranking functions seem to be used rather sparingly in computer science which we attribute to the difficulties that arise from the requirement of enumerating all world models. By presenting an example application which bypasses these difficulties we hope to encourage further applications of ranking functions. Since the presented example is still rather artificial, we nevertheless believe that it captures important difficulties present in any computer vision problem. The obvious direction for future research is hence to actually develop real computer vision applications in which a ranking function contributes significantly. For instance, in the process of developing the presented work, we applied the two-level-learning architecture to a (still simulated) task of object recognition, where an agent had to distinguish between similar three-dimensional objects [11]. There, the ability to use solely features in the state description also removed the need for establishing a global co-ordinate system.

Acknowledgments.

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

References

1. Alchourron, C.E., Gardenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. *J. Symbolic Logic*, **50**(2): 510–530 (1985)
2. Bayes, T.: An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London*, **53**: 370–418 (1763)
3. Blockeel, H., De Raedt, L.: *Top-Down Induction of First-Order Logical Decision Trees.* Artificial Intelligence, **101**: 285–297 (1998)
4. Caruana, R., Niculescu-Mizil, A.: *An empirical comparison of supervised learning algorithms.* Proceedings of the 23rd international conference on Machine learning: 161–168 (2006)
5. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. *Artificial Intelligence*, **89**: 1–29 (1996)
6. Driessens, K., Ramon, J.: *Relational instance based regression for relational reinforcement learning.* Proceedings of the Twentieth International Conference on Machine Learning: 123–130 (2003)
7. Dzeroski, S., Raedt, L.D., Driessens, K.: *Relational reinforcement learning.* Machine Learning. vol. 43: 7–52 (2001)
8. Gartner, T., Driessens, K., Ramon, J.: *Graph kernels and gaussian processes for relational reinforcement learning.* Inductive Logic Programming, 13th International Conference, ILP (2003)
9. Gombert, J.E.: Implicit and explicit learning to read: Implication as for subtypes of dyslexia. *Current psychology letters*, **1**(10) (2003)

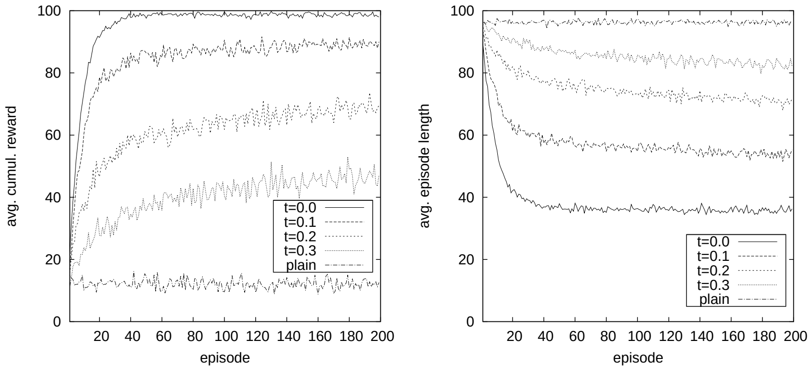

Figure 4: Learning progress using a feature detection probability of 0.5.

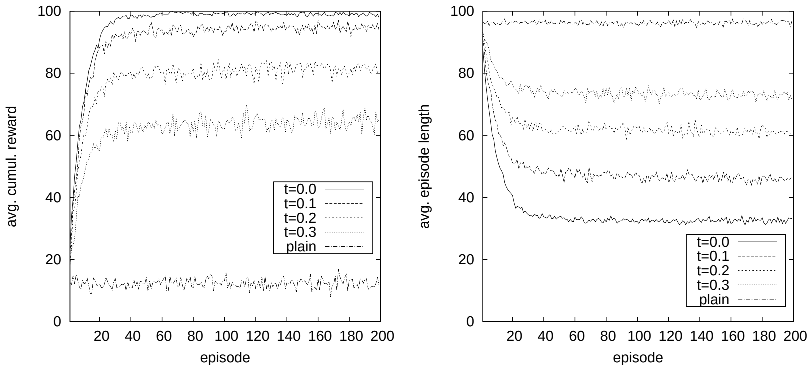

Figure 5: Learning progress using a feature detection probability of 0.9.

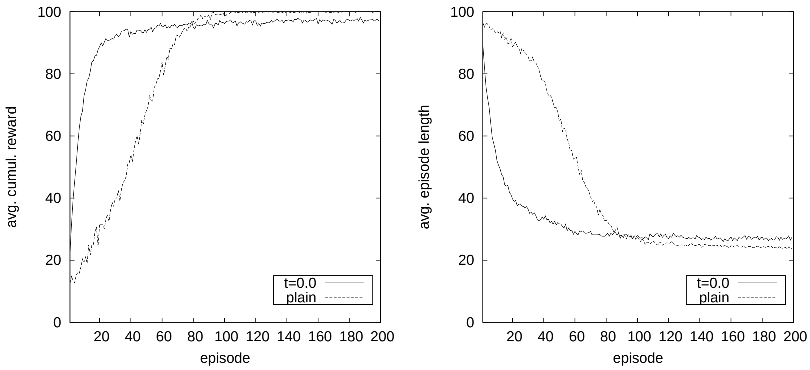

Figure 6: Learning progress using a feature detection probability of 1.0.

10. Häming, K., Peters, G.: *An alternative approach to the revision of ordinal conditional functions in the context of multi-valued logic.* 20th International Conference on Artificial Neural Networks: 200–203. Springer-Verlag, Greece (2010)

11. Häming, K., Peters, G.: *A hybrid learning system for object recognition.* 8th International Conference on Informatics in Control, Automation, and Robotics (ICINCO 2011). Noordwijkerhout, The Netherlands (2011)

12. Häming, K., Peters, G.: *Improved revision of ranking functions for the generalization of belief in the context of unobserved variables.* International Conference on Neural Computation Theory and Applications (2011)

13. Jaccard, P.: Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, **37**: 547–579 (1901)

14. Kern-Isberner, G.: *Postulates for conditional belief revision.* IJCAI'99: Proceedings of the 16th international joint conference on Artifical intelligence: 186–191. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)

15. Leopold, T., Kern Isberner, G., Peters, G.: *Combining reinforcement learning and belief revision: A learning system for active vision.* 19th British Machine Vision Conference (BMVC 2008). **1**: 473–482. Leeds, UK (2008)

16. Reber, AS.: Implicit learning and tacit knowledge. *Journal of Experimental Psycology: General*, **3**(118): 219–235 (1989)

17. Spohn, W.: Ordinal conditional functions: A dynamic theory of epistemic states. *Causation in Decision, Belief Change and Statistics*: 105–134 (1988)

18. Spohn, W.: Ranking functions, agm style. *Internet Festschrift for Peter Gärdenfors* (1999)

19. Spohn, W.: Causation: An alternative. *British Journal for the Philosophy of Science*, **57**(1): 93–119 (2006)

20. Spohn, W.: A survey of ranking theory. *Degrees of Belief*. Springer (2009)

21. Sun, R., Merrill, E., Peterson, T.: From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*. **25**: 203–244 (2001)

22. Sun, R.: Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence*, **75**: 241–295 (1995)

23. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge (1998)

24. Wu, H.C., Luk, R.W.P., Wong, K.F., Kwok, K.L.: Interpreting tf-idf term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, **26**: 13:1–13:37 (2008)

25. Xu, Y., Fern, A., Yoon, S.: Learning linear ranking functions for beam search with application to planning. *J. Mach. Learn. Res.*, **10**: 1571–1610 (2009)