Ranking Functions in Large State Spaces

Klaus Häming, Gabriele Peters

Ranking Functions in Large State Spaces

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. Large state spaces pose a serious problem in many learning applications. This paper discusses a number of issues that arise when ranking functions are applied to such a domain. Since these functions, in their original introduction, need to store every possible world model, it seems obvious that they are applicable to small toy problems only. To disprove this we address a number of these issues and furthermore describe an application that indeed has a large state space. It is shown that an agent is *enabled* to learn in this environment by representing its belief state with a ranking function. This is achieved by introducing a new entailment operator that accounts for similarities in the state description.

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

1 Introduction

This paper discusses the applicability of ranking functions on problem domains which have a large state space. We discuss this issue with an autonomous agent in mind whose belief state is represented as such a function. This agent has to incorporate its experience into its belief state in order to learn and solve its task successfully. The considerations in this paper are assessed by implementing such an agent and expose him to a suitable task. This is done in an reinforcement learning [10] application, which has been augmented by a two-level representation of the agents belief. This two-level representation has been inspired by the work of Sun [9, 8] and is based on psychological findings, e.g., of Reber [5] and Gombert [3]. For the largest part however, this work discusses ranking functions independently from a concrete application.

Ranking functions were introduced by Spohn [6] under the term of ordinal conditional functions. They were introduced to account for the dynamics of belief revision [1, 2]. 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 [7]. Ranking functions 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 an elegant incorporation of conditional information.

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 [11] which is often used in text mining. These ranking functions are a tool to *cope* with large state spaces, as shown in, e.g., [12]. 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.

The next section provides a brief review of Spohn's ranking functions, followed by a discussion of their problems with large state spaces. A proposal of how to handle this is given thereafter. Since uncertainty is also a problem for belief representations, we discuss this in Section 5 with our example application in mind, 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. 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. The agent believes in a model M, iff

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

Besides querying the rank of a model, we may ask for the rank of a formula F. 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 $\min\{\kappa(M) | M \models F\}$. So, if the agent believes in a particular world model which happens to entail F, it will also belief F.

Whenever the agent experiences new information about its world, new ranks are assigned to the models to reflect the agent's new belief state. This incorporation of new information is called *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 $\alpha \in \mathbb{N}$, which is enforced during revision.

We will 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, the revision $\kappa * (P, \alpha)$ is given by*

$$\kappa * (P,\alpha)(M) = \begin{cases} \kappa(M) & : \kappa(\overline{P}) \geq \alpha \\ \kappa(M) - \kappa(P) & : \kappa(\overline{P}) < \alpha \wedge M \models P \\ \kappa(M) + \alpha - \kappa(\overline{P}) & : \kappa(\overline{P}) < \alpha \wedge M \models \overline{P} \end{cases} \tag{2}$$

Essentially this definition states that

1. there is nothing to do, if P is already believed with strength α
2. otherwise two partitions of the set of models are relevant:
   (a) those, that agree on P, and whose ranks are modified so that we assign rank 0 to the least disbelieved model.
   (b) those, that agree on $\overline{P}$, and whose ranks are modified so that $\kappa(\overline{P}) \geq \alpha$ holds afterwards.

The rationale behind this partitioning is that the models in $\{M|M \models P\}$ are independent *conditional* on P and should therefore keep their *relative* ranks. The same argument is applied to $\{M|M \models \overline{P}\}$ and $\overline{P}$.

After this 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. As shown in [4], it is advisable to use the additional operator $\kappa[F] := \max\{\kappa(M)|M \models F\}$ to define this revision.

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

$$\kappa * (B|A,\alpha)(M) := \begin{cases} \kappa(M) & : D \geq \alpha \\ \kappa(M) - \kappa(A \Rightarrow B) & : D < \alpha \wedge M \models (A \Rightarrow B) \\ \kappa[AB] - \kappa(A \Rightarrow B) + \alpha + \kappa(M) - \kappa(A\overline{B}) & : D < \alpha \wedge M \models (A\overline{B}) \end{cases} \tag{3}$$

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 seemingly more complex rules of Definition 2 emerge from the fact, that $\{M|M \models (A \Rightarrow B)\} = \{M|M \models \overline{A}\} \cup \{M|M \models AB\}$, while the postcondition $\kappa(A\overline{B}) - \kappa[AB] \geq \alpha$ only refers to $\kappa(A\overline{B})$ and $\kappa(AB)$.

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. 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 for clarity.

$$\begin{array}{c|c|c} \kappa & \xrightarrow{\kappa*((a=2)\Rightarrow(c=2),1)} & \kappa' \\ \hline \underline{2\,1\,1} & & \underline{2\,1\,2} \\ 2\,1\,2 & & \\ 2\,2\,1 & & 2\,2\,2 \\ 2\,2\,2 & \rightarrow & 2\,1\,1 \\ & & \\ & & 2\,2\,1 \\ \vdots & & \vdots \end{array} \tag{4}$$

The right column of this example captures the belief after κ has been revised with $(a = 2) \Rightarrow (c = 2)$ using Definition 2. 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 $\kappa(A\overline{B}) - \kappa[AB] = 1$.

3 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:

$$\begin{array}{c} \kappa \\ \hline M_1\ M_2\ M_3 \cdots \lightning \\ M_4\ M_5 \\ \vdots \\ \lightning \end{array}$$

We may try to circumvent this problem by modeling a skeptic agent, one that initially disbelieves anything. We may think that this means we commit to a scenario, in which initially all models have rank ∞ and simply omit those. That would mean our initial ranking function is empty.

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

$$\kappa(M) = 0 \vee \kappa(\overline{M}) = 0\ . \tag{5}$$

If we are to put every single model initially to rank ∞, this rule will be violated. Fortunately, the actual rank is less important that 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

$$\kappa(M) = 0 \wedge \kappa(\overline{M}) = 0 \,, \tag{6}$$

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 $\kappa(\overline{M}) = \infty$. Hence, a skeptic agent indeed allows us to omit not yet experienced world models.

Concerning revision in large state spaces, we 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 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 $A \Rightarrow B$, therefore the models $\{M|M \models \overline{A} \vee B\}$ have to be shifted towards rank 0. If we re-consider the example of the 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 ranking function have to be created to maintain their relative ranks. Furthermore, the operator $\kappa[AB]$ creates additional complications. If $\kappa[AB] = \infty$, we will also have to create all $\{M|M \models AB \ \wedge \kappa(M) = \infty\}$.

4 The Proposed Solution

If we want to use ranking functions on larger 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 partitions 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, the postcondition $\kappa(\overline{P}) \geq \alpha$ must hold afterwards. Just taking P and those models into account that have been experienced so far obviously complies with this requirement. This is true, because either there are models from $\{M|M \models \overline{P}\}$ 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 not such models, there is nothing to do because the postcondition already holds.

Similarly, for a revision with conditional information, we want the postcondition $\kappa(A\overline{B}) - \kappa[AB] \geq \alpha$ to hold afterwards. So, if κ is to be revised with $A \Rightarrow B$ and $\kappa(\overline{A} \vee B) = \infty$, then we may shift $\{M|M \models A \wedge B\}$ 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

$$\kappa[AB] = \max\{\kappa(M) | M \models AB \wedge \kappa(M) < \infty\} , \tag{7}$$

i.e., it returns the highest ranked *known* model of AB. Fortunately, this still leads to revisions after which the conditional $(B|A)$ 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.

5 Partially Known or Noisy State Spaces

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. 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 from the same position may therefore yield slightly different descriptions of the visual input.

Assume the state description takes the form

$$S = v_1 \wedge v_2 \wedge v_3 \wedge v_4 \wedge \ldots v_n , \tag{8}$$

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. This is the case, if the possible features are not known beforehand but added as they appear:

$$\begin{array}{c} \kappa \\ \hline \left\| \begin{array}{c} M_1 = v_1 \wedge \overline{v_2} \wedge v_3 \wedge v_4 \wedge \overline{v_5} \wedge \cdots \not{\ell} \\ \vdots \end{array} \right\| \end{array}$$

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

In the 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 conditional $A \Rightarrow B$ we are able to partition the models of $\mathfrak{W}$ into the three subsets $\{M | M \models \overline{A}\}$, $\{M | M \models AB\}$, and $\{M | M \models A\overline{B}\}$. This is required to apply Definition 2. To enforce this, we require the conditionals to be elements of the following set $\mathfrak{C}$:

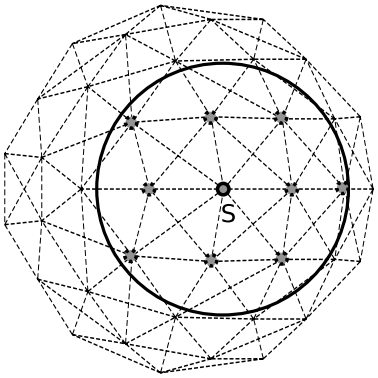

Fig. 1. Simulated Features. Assume, that the current state is S. Then, a visual perception can be simulated by using the grid nodes in the vicinity of S.

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

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

6 Example Application

To assess the applicability of the aforementioned ideas, we use an experimental set-up which presents an environment to the agent that is perceived as a set of (unreliable) features as introduced in Section 5.

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 on the grid, includeing 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.

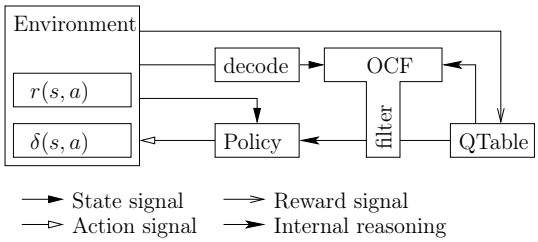

Fig. 2. Two-level reinforcement learning set-up. The ranking function ("OCF") acts as a filter on the possible actions a reinforcement learning agent may take.

The agent uses the information in its state signal about visible features to determine its position above the object. Fig. 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.

We used the two-level-learning approach of [4], in which a ranking function acts as a filter on the possible actions a reinforcement learning [10] agent may take. Fig. 2 clarifies the architecture. There, δ is a transition function and r a reward function. The agent explores an environment through trial and error until a goal state is reached. The arrival at the goal state triggers a reward of 100. All other actions are neither rewarded nor punished. The policy is an ϵ-greedy policy with $\epsilon = 0.1$. Fig. 2 also shows the presence a "decode" module. This module creates the symbolical state description from the state signal the environment provides. The Q-table is realized by applying a hash-function on the state description to create a suitable integer number.

The conditionals used for revision are created with the aid of this Q-table. In each state S a revision with the conditional $S \Rightarrow A$ takes place, if a best action A has been identified by the Q-table.

As shown in Fig. 3, the answer to the question asked in Section 5 about a good value of 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. Requiring it to recognize more features hinders its learning and therefore slows its progress. For comparison, we also included the graph of a simple Q-learner into Fig. 3 ("plain"). This Q-learner uses a Q-table without a supplementing ranking function. It is not able to show any progress at all.

7 Conclusion

This work discusses the applicability of ranking functions in situations where large state spaces seem to strongly discourage their usage. A number of potential problems such as the instantiation of a large number of models during revision and uncertainty in the state description have been discussed. ranking

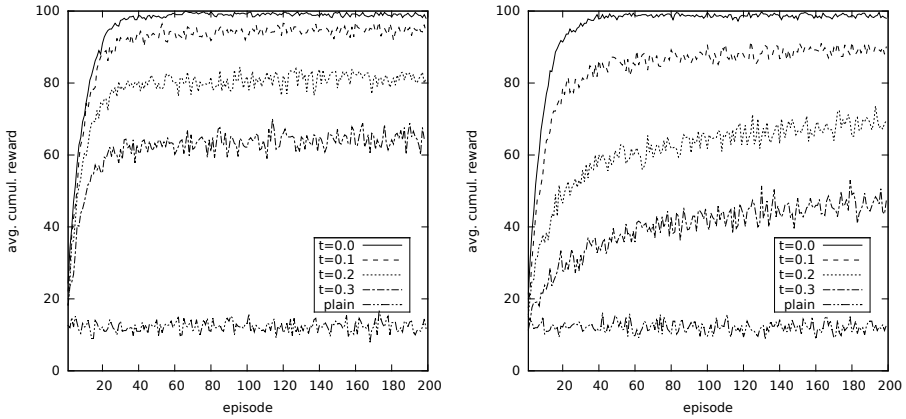

Fig. 3. Learning progress over 200 episodes, averaged over 500 runs. The graphs of the left figure have been calculated with a feature detection probability of 0.9 (cf. Fig. 1 and text), while the right figure shows the same experiments for a detection probability of 0.5.

functions seem to be used rather sparingly which we attribute to these difficulties. To encourage their application we presented an example application from the domain of reinforcement learning. In this example, ranking functions allow a learner to succeed where a traditional Q-learner fails. This has been achieved by introducing a new entailment operator that accounts for similarities in the state description.

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. Darwiche, A., Pearl, J.: On the Logic of Iterated Belief Revision. Artificial intelligence 89, 1–29 (1996)
3. Gombert, J.E.: Implicit and Explicit Learning to Read : Implication as for Subtypes of Dyslexia. Current psychology letters 1(10) (2003)
4. Häming, K., Peters, G.: An Alternative Approach to the Revision of Ordinal Conditional Functions in the Context of Multi-Valued Logic. In: 20th International Conference on Artificial Neural Networks. Thessaloniki, Greece (2010)
5. Reber, A.S.: Implicit Learning and Tacit Knowledge. Journal of Experimental Psycology: General 3(118), 219–235 (1989)
6. Spohn, W.: Ordinal Conditional Functions: A Dynamic Theory of Epistemic States. Causation in Decision, Belief Change and Statistics pp. 105–134 (August 1988)

7. Spohn, W.: Ranking Functions, AGM Style. Internet Festschrift for Peter Gärdenfors (1999)
8. Sun, R., Merrill, E., Peterson, T.: From Implicit Skills to Explicit Knowledge: A Bottom-Up Model of Skill Learning. In: Cognitive Science. vol. 25, pp. 203–244 (2001)
9. Sun, R.: Robust Reasoning: Integrating Rule-Based and Similarity-Based Reasoning. Artif. Intell. 75, 241–295 (1995)
10. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
11. Wu, H.C. and Luk, R.W.P. and Wong, K.F. and Kwok, K.L.: Interpreting TF-IDF Term Weights as Making Relevance Decisions. ACM Trans. Inf. Syst. 26, pp. 13:1–13:37 (2008)
12. Xu, Y. and Fern, A. and Yoon, S.: Learning Linear Ranking Functions for Beam Search with Application to Planning. J. Mach. Learn. Res. 10, pp. 1571–1610 (2009)