*ECAI 2008*
*M. Ghallab et al. (Eds.)*
*IOS Press, 2008*

*doi:10.3233/978-1-58603-891-5-65*

# Belief revision with reinforcement learning for interactive object recognition

**Thomas Leopold**[1] and **Gabriele Kern-Isberner**[2] and **Gabriele Peters**[3]

**Abstract.** From a conceptual point of view, belief revision and learning are quite similar. Both methods change the belief state of an intelligent agent by processing incoming information. However, for learning, the focus in on the exploitation of data to extract and assimilate useful knowledge, whereas belief revision is more concerned with the adaption of prior beliefs to new information for the purpose of reasoning. In this paper, we propose a hybrid learning method called SPHINX that combines low-level, non-cognitive reinforcement learning with high-level epistemic belief revision, similar to human learning. The former represents knowledge in a sub-symbolic, numerical way, while the latter is based on symbolic, non-monotonic logics and allows reasoning. Beyond the theoretical appeal of linking methods of very different disciplines of artificial intelligence, we will illustrate the usefulness of our approach by employing SPHINX in the area of computer vision for object recognition tasks. The SPHINX agent interacts with its environment by rotating objects depending on past experiences and newly acquired generic knowledge to choose those views which are most advantageous for recognition.

## 1 INTRODUCTION

One of the most challenging tasks of computer vision systems is the recognition of known and unknown objects. An elegant way to achieve this is to show the system some samples of each object class and thereby train the system, so that it can recognize objects that it has not seen before, but which look similar to some objects of the training phase (due to some defined features).

Several methods to do so have been successfully used and anaylized. One of them is to set up a rule-based system and have it reason, another one is to use numerical learning methods such as reinforcement learning. Both of them have advantages, but also disadvantages. Reinforcement learning yields good results in different kinds of environments, but its training is time consuming, since it is a trial-and-error method and the agent has to learn from scratch. The possibilities to introduce background knowledge (e. g., by the choice of the initial values of the QTable) are more limited as for example with knowledge representation techniques. Another disadvantage consists in a limited possibility to generalize experiences and so to be able to act appropriately in unfamiliar situations. Though some generalization can be obtained by the application of function approximization techniques, the possibilities to generalize from learned rules to unfamiliar situations are more diverse again with for example knowledge representation techniques.

Knowledge representation and belief revision techniques have the advantage that the belief of the agent is represented quite clearly and allows reasoning about actions. The belief can be extended by new information, but needs to be revised when the new information contradicts the current belief. One drawback is that it is difficult to decide which parts of the belief should be given up, so that the new belief state is consistent, i.e., without inherent contradictions.

In this paper, we present our hybrid learning system SPHINX, named after the Egyptian statue of a hybrid between a human and a lion. It combines the advantages of both Q-Learning and belief revision and diminishes the disadvantages, thus synergy effects can emerge. SPHINX agents, on the one hand, are intelligent agents equipped with epistemic belief states which allows them to build a model of the world and to apply reasoning techniques to focus on most plausible actions. On the other hand, they use QTables to determine which action should be carried out next, and are able to process reward signals from the environment. Moreover, SPHINX agents can learn situational as well as generic knowledge which is incorporated into their epistemic states via belief revision. In this way, they are able to adjust faster and more thoroughly to the environment, and to improve their learning capabilites considerably. This will be illustrated in detail by experiments in the field of computer vision.

This paper is organized as follows: Chapter 2 summarizes related work. In chapter 3 we recall basic facts on Q-Learning, ordinal conditional functions and revision. Chapter 4 contains the main contribution of this paper, the presentation of the SPHINX system. Chapter 5 summarizes results from experiments in computer vision carried out in different environments. Finally, we conclude in chapter 6.

## 2 RELATED WORK

Psychological findings propose a two-level learning model for human learning [1], [6], [3], [10]. On the so called bottom level, humans learn *implicitly* and acquire *procedural* knowledge. They are *not aware* of the relations they have learned and can hardly put it into words. On the other level, the top level, humans learn *explicitly* and acquire *declarative* knowledge. They are *aware* of the relations they have learned and can express it, e. g., in form of if-then rules. A special form of declarative knowledge is *episodic* knowledge. This kind of knowledge is not of general nature, but refers to *specific* events, situations or objects. Episodic knowledge makes it possible to remember specific situations where general rules do not hold.

These two levels do not work separately. Depending on what is learned, humans learn top-down or bottom-up [11]. It has been found [8] that in completely unfamiliar situations mainly implicit learning

[1] University of Technology Dortmund, Germany, email: thleopold@hotmail.com

[2] University of Technology Dortmund, Germany, email: gabriele.kern-isberner@cs.uni-dortmund.de

[3] University of Applied Sciences and Arts Dortmund, Germany, email: gabriele.peters@fh-dortmund.de

takes place and procedural knowledge is acquired. The declarative knowledge is formed afterwards. This indicates that the bottom-up direction plays an important role. It is also advantageous to continually verbalize to a certain extent what one has just learned and so speed up the acquisition of declarative knowledge and thereby the whole learning process.

Sun, Merrill and Peterson developed the learning model CLARION [9]. It is a two-level, bottom-up learning model which uses Q-Learning for the bottom level and a set of rules for the top level. The rules have the form 'Premise ⇒ Action', where the premise can be met by the current state signal of the environment. For the maintainance of the set of rules (i. e., adding, changing and deleting rules) the authors have conceived a certain technique. They have proven their model, which works similar to human learning, to be successful in a mine field navigation task and similar to human learning.

Cang Ye, N. H. C. Yung and Danwei Wang propose a neural fuzzy system [2]. Like CLARION, this is a two-level learning model, combining reinforcement learning and fuzzy logic. The system has successfully been applied to a mobile robot navigation task.

## 3 BASICS AND BACKGROUND

In this section, we will recall basic facts on the two methodologies that are used and combined in this paper.

First, we briefly describe Q-Learning, a popular approach used for solving Markov Decision Processes (MDPs) (see e.g. [12]). The scenario is the usual one for agents, where one or more agents interact with an environment. Normally, the environment starts in a state and ends, when one terminal state is reached. This timespan is called an episode. For each action, the agent is rewarded. The more reward it collects during an episode, the better. Episodes consist of steps in which the agent first perceives the current state $s$ of the environment via a (numerical) state signal, e. g., an ID. It looks up in its memory, called QTable, which action $a$ seems to be the best in this situation and performs it. The environment reacts on this action by changing its state to $s'$. After this change, the agent gets a reward $r$ for its choice and updates its QTable.

$Q(\lambda)$-learning is an enhanced Q-Learning method that not only takes the expected rewards into account but also considers the state-action-pairs that have led to a state $s$. Let $Q(s, a)$ represent the sum of rewards the agent expects to receive until the end of the episode, if it performs action $a$ in situation $s$, and let $A(s)$ be the set of actions the agent can perform in state s. The update formula for a state-action-pair $(\tilde{s}, \tilde{a})$ for $Q(\lambda)$-learning is $Q(\tilde{s}, \tilde{a}) := Q(\tilde{s}, \tilde{a}) + \alpha \cdot e(\tilde{s}, \tilde{a}) \cdot \delta$, where $e(\tilde{s}, \tilde{a})$ is an eligibility factor, expressing how much influence on $(s, a)$ is conceded to $(\tilde{s}, \tilde{a})$ (the longer ago, the smaller the value), and $\delta := r + \max\limits_{a' \in A(s')} Q(s', a') - Q(s, a)$. Before updating the $(\tilde{s}, \tilde{a})$-values, the eligibility factor of the current state-action-pair $(s, a)$ is increased by 1. After the update, the parameter $\lambda$ is used to decrease the $e(\tilde{s}, \tilde{a})$-values to $e(\tilde{s}, \tilde{a}) := \lambda \cdot e(\tilde{s}, \tilde{a})$. For $\lambda = 0$, we get the basic Q-Learning approach.

The decision which action to take in a situation $s$ is usually done by choosing the one with the greatest $Q(s, a)$-value. To make the discovery of new solutions possible, the agent chooses a random action with a small probability $\epsilon$.

Now, the concept of ordinal conditional functions (OCFs) and appropriate revision techniques will be explained. OCFs will serve as representations of epistemic states of agents in this paper. Ordinal conditional functions [7] are also called ranking functions, as they assign a degree of plausibility in the form of a degree of disbelief, or surprise, respectively, to each possible world. We will work within a propositional framework, making use of multi-valued propositional variables $d_i$ with domains $\{v_{i,1}, \ldots, v_{i,m_i}\}$. Possible worlds are simply interpretations here, assigning exactly one value to each $d_i$, and thus correspond to complete elementary conjunctions of multivalued literals $(d_i = v_{i,j})$, mentioning each $d_i$. Let $\Omega$ be the set of all possible worlds. Formally, an *ordinal conditional function (OCF)* is a mapping $\kappa : \Omega \to \mathbb{N} \cup \{\infty\}$ with $\kappa^{-1}(0) \neq \emptyset$. The lower $\kappa(\omega)$, the more plausible is $\omega$, hence the most plausible worlds have $\kappa$-value 0. A degree of plausibility can be assigned to formulas $A$ by setting $\kappa(A) := \min\{\kappa(\omega) \mid \omega \models A\}$, so that $\kappa(A \vee B) = \min\{\kappa(A), \kappa(B)\}$. This means that a formula is considered as plausible as its most plausible models. Therefore, due to $\kappa^{-1}(0) \neq \emptyset$, at least one of $\kappa(A), \kappa(\overline{A})$ must be 0. A proposition $A$ is believed if $\kappa(\overline{A}) > 0$ (which implies particularly $\kappa(A) = 0$). Moreover, degrees of plausibility can also be assigned to conditionals by setting $\kappa(B|A) = \kappa(AB) - \kappa(A)$. A conditional $(B|A)$ is accepted in the epistemic state represented by $\kappa$, or $\kappa$ *satisfies* $(B|A)$, written as $\kappa \models (B|A)$, iff $\kappa(AB) < \kappa(A\overline{B})$, i.e. iff $AB$ is more plausible than $A\overline{B}$.

OCFs represent the epistemic attitudes of agents in quite a comprehensible way and offer simple arithmetics to propagate information. Therefore, they can be revised by new information in a straightforward manner, making use of the idea of so-called *c-revisions* [4] that are capable of revising ranking functions even by sets of new conditional beliefs. Here, we will only consider revisions by one conditional belief, so we will present the technique for this particular case.

Given a prior epistemic state in the form of an OCF $\kappa$ and a new conditional belief $(B|A)$, the revision $\kappa^* = \kappa * (B|A)$ is defined by

$$\kappa^*(\omega) = \begin{cases} \kappa_0 + \kappa(\omega) + \lambda, & \text{if } \omega \models A\overline{B}, \\ \kappa_0 + \kappa(\omega) \quad\quad , & \text{otherwise}, \end{cases} \tag{1}$$

where $\kappa_0$ is a normalizing additive constant and $\lambda$ is the least natural number to ensure that $\kappa^*(AB) < \kappa^*(A\overline{B})$. Although c-revisions are defined in [4] for logical languages defined from binary atoms, the approach can be easily generalized to considering multi-valued propositional variables. Note that also c-revision by facts is covered, as facts are identified with degenerate conditionals with tautological premises, i.e. $A \equiv (A|\top)$.

OCFs and c-revisions provide a framework to carry out high quality belief revision meeting all standards which are known to date, even going beyond that [4].

## 4 THE SPHINX LEARNING METHOD

Similar to the cognitive model, our learning method consists of two levels. For the bottom level we use $Q(\lambda)$-Learning, and for the top level, ordinal conditional functions (OCFs) are employed to represent the epistemic state of an agent and perform belief revision. This brings together two powerful methodologies from rather opposite ends of the scale of cognitive complexity, meeting the challenge of combining learning and belief revision in a particularly extreme case.

To combine belief revision and reinforcement learning, each (subsymbolic) state $s$ is described by a logical formula from a language defined over propositional variables $d_i$ with domains $\{v_{i,1}, \ldots, v_{i,m_i}\}$. The symbolic representation of a specific state is a conjunction of literals mentioning all $d_i$ and reflects the logical perception of $s$ by the agent. Furthermore, we define a variable $action$ having as domain the set $Actions$ of possible actions. Hence, the possible worlds on which ranking functions are defined here correspond to elementary conjunctions of the form $(d_1 = v_{1,k_1}) \wedge \ldots \wedge (d_n = v_{n,k_n}) \wedge (action = a)$.

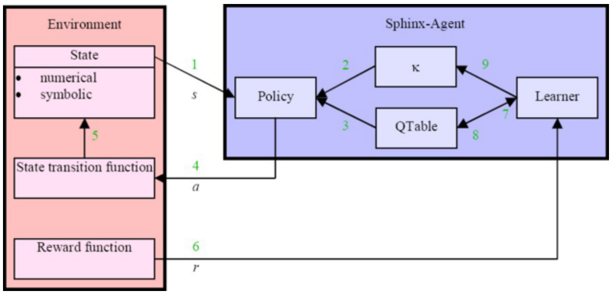

**Figure 1.** The SPHINX system

The SPHINX system interlinks Q-learning, the epistemic state and belief revision in two ways: First, it uses current beliefs to restrict the search space of actions for Q-Learning. Second, direct feedback to an action in the form of a reward is processed to acquire specific or generic symbolic knowledge from the most recent experience by which the current epistemic state is revised. It is displayed in figure 1 and works as follows:

**Algorithm 'Sphinx-Learning':**
*While* the current state $s$ is not a terminal state

1. The Sphinx agent perceives the signal of the state $s$ coming from the environment and its logical description $d(s)$.
2. The agent queries its current epistemic state $\kappa$ which actions $A_\kappa(s) = \{a_1, \ldots, a_k\}$ are most plausible in $s$.
3. The agent looks up the Q-values of these actions and determines the set $A_{best}(s) \subseteq A_\kappa(s)$ of those actions in $A_\kappa(s)$ that have the greatest Q-value.
4. The agent chooses a random action $a \in A_{best}(s)$ and performs it.
5. The environment changes to the successor state.
6. The agent receives the reward $r$ from the environment.
7. The agent updates the QTable as described in section 3.
8. The new Q-values for actions in $s$ are being read and the new best actions for $s$ are determined.
9. The agent tries to find new rules that relate $d(s)$ to best actions (according to the updated QTable) and revises $\kappa$ with this information in form of conditionals.

*End While*

We will now explain the algorithm step by step. When a state $s$ is perceived (step 1), then $\kappa$ is browsed for the most plausible worlds satisfying $d(s)$. $A_\kappa(s)$ (step 2) is the set of actions occurring in the most plausible $d(s)$-worlds:

$$A_\kappa(s) = \{a \in \mathit{Actions} \mid \kappa(d(s) \wedge \mathit{action} = a) = \kappa(d(s))\}$$

Then, the actions in $A_\kappa(s)$ are filtered according to their Q-values (step 3), and one of these actions is carried out (step 4). It is particularly in these two steps that the enhancement of reinforcement learning with epistemic background pays out, since an ordinary Q-Agent determines the set of best actions from the set of *all* possible actions. Steps 5 to 7 are pure Q-Learning.

In step 8, the best actions for $s$ due to the new Q-values are determined. This is done to exploit the experience by the received reward for future situations and make it usable on the epistemic level in step 9. The operations performed in step 9 are quite complex and described in the following. The aim of the mentioned revision of $\kappa$ is to make those actions most plausible in $d(s)$ that have the greatest Q-value in $s$. As inputs for this revision, the agent tries to find patterns in the state descriptions for which certain actions are generally better than others. This is done by a frequency based heuristics. For each pattern (i.e., a conjunction of literals of *some* of the variables) $p$ and each action $a$, the agent remembers how often $a$ was a best resp. a poor action by using counters. If the agent finds in step 8, that an action $a$ is a best action in $s$ and has not been among the best actions before, then the counters for $a$ of all patterns covered by $d(s)$ are increased by 1. If $a$ was a best action in $s$ before but is no longer, the counters are decreased by 1. Negative experiences where $a$ was a poor action are handled in an analogous manner. With these counters, probabilities can be calculated, expressing, if $a$ is usually a best resp. a poor action, when a situation $s$ for which $d(s)$ satisfies $p$ is perceived.

If such a relation between a pattern and a set of actions is found, a revision of $\kappa$ with a conditional encoding such newly acquired strategic knowledge is performed; basically, the following four different types of revision occur:

1. Revision with information about a poor action in a specific state (episodic knowledge).
2. Revision with information about a poor action in several, similar states (generalization).
3. Revision with information about best actions in a specific state (episodic knowledge).
4. Revision with information about best actions in several, similar states (generalization).

A 'poor' action in a specific state resp. in several, similar states was defined as an action that yields a reward less than -1. The conditionals used to revise $\kappa$ have the following forms:

1. $(\overline{\mathit{action} = a} | d(s))$, where $d(s)$ is the symbolic representation of a certain state $s$ in which $a$ is poor.
2. $(\overline{\mathit{action} = a} | p)$, where $p$ is a pattern satisfied by $d(s)$, representing a set of states, which are similar because they share a common pattern.
3. $(\bigvee_i \mathit{action} = a_i | d(s))$, where all $a_i$ are best actions (due to their Q-values) in $s$.
4. $(\bigvee_i \mathit{action} = a_i | p)$, where each $a_i$ is a best action in at least *one* of the states covered by the pattern $p$. $a_i$ needs not to be a best action in *all* states covered by $p$.

The last form of revision should exclude not best actions from being plausible when $p$ is perceived, so the agent has to find the best action for a specific state covered by $p$ only among the actions $a_i$.

Since revisions and especially revisions with generalized rules have a strong influence on the choice of actions, they have to be handled carefully, i. e., the agent should be quite sure about the correctness of a rule before adding it to its belief. Therefore, the agent uses several counters counting, how often an action has been poor, not poor, a best or not a best one under certain circumstances. With these counters some probabilities can be calculated which can be used to evaluate the certainty about the correctness of a specific rule. However, since all rules are merely plausible but not correct in a logical sense, further revisions may alleviate or even cancel the effects of erroneously acquired rules.

Our learning model also supports background knowledge. If the user knows some rules that might be helpful for the agent and its task, he can formulate them as conditionals and let the agent revise $\kappa$ with them before starting to learn.

## 5 INTERACTIVE OBJECT RECOGNITION

We tested our learning method in a navigation environment and in two different simulations of object recognition environments. In this paper, we present the results of the latter in two different scenarios.

## 5.1 Recognition of Geometric Objects

In this test environment, the agent has to learn to recognize the following objects: sphere, ellipsoid, cylinder, cone, tetrahedron, pyramid, prism, cube, cuboid. By interacting with the environment the agent can look at the object from the front, from the side or from the top or it can choose to try to name the current object. The possible front, side, and top views are represented by five elementary shapes, namely: circle, ellipse, triangle, square, and rectangle. For example, the cone has the front view 'triangle', the side view 'triangle', and the top view 'circle'. The prism is given by the front view 'triangle', the side view 'rectangle', and the top view 'rectangle'. This leads to the following domains for this environment:

- $\mathit{FrontView} = \{\mathit{Unknown}, \mathit{Circle}, \mathit{Ellipse}, \mathit{Triangle}, \mathit{Square}, \mathit{Rectangle}\}$
- $\mathit{SideView} = \{\mathit{Unknown}, \mathit{Circle}, \mathit{Ellipse}, \mathit{Triangle}, \mathit{Square}, \mathit{Rectangle}\}$
- $\mathit{TopView} = \{\mathit{Unknown}, \mathit{Circle}, \mathit{Ellipse}, \mathit{Triangle}, \mathit{Square}, \mathit{Rectangle}\}$
- $\mathit{Action} = \{\mathit{LookAtFront}, \mathit{LookAtSide}, \mathit{LookAtTop}, \mathit{RecognizeUnknown}, \mathit{RecognizeSphere}, \mathit{RecognizeEllipsoid}, \mathit{RecognizeCylinder}, \mathit{RecognizeCone}, \mathit{RecognizeTetrahedron}, \mathit{RecognizePyramid}, \mathit{RecognizePrism}, \mathit{RecognizeCube}, \mathit{RecognizeCuboid}\}$

At the beginning of each episode, the environment chooses one of the nine geometric objects and generates the state signal '$\mathit{FrontView} = \mathit{Unknown} \wedge \mathit{SideView} = \mathit{Unknown} \wedge \mathit{TopView} = \mathit{Unknown}$'. If the agent's action is *LookAtFront*, *LookAtSide*, resp. *LookAtTop*, the *FrontView*, *SideView*, resp. *TopView* is revealed in the new state signal following the agent's action. If the agent's action is an action of type '*Recognize*' action, the episode ends.

The reward function returns -1, if one of the '*Look*' actions has been performed. Otherwise, the agent is rewarded with 10, if it has recognized the objects correctly, and with -10, if not. After ten steps the running episode is forced to end. Figure 2 shows the recognition rates after each training phase. In each training phase, each object is shown ten times to the current agent. The values result from 1000 independend agents.

If the agents are provided with the background knowledge *If no view has been perceived yet, then look at the front, the side, or the top of the object* via the conditional $(\mathit{action} = \mathit{LookAtFront} \vee \mathit{action} = \mathit{LookAtSide} \vee \mathit{action} = \mathit{LookAtTop} | \mathit{FrontView} = \mathit{Unknown} \wedge \mathit{SideView} = \mathit{Unknown} \wedge \mathit{TopView} = \mathit{Unknown})$, the recognition rates improve, as can also be seen from figure 2.

In the following, we list some of the rules that the agents learned by exploring the effects of updating the QTables on the cognitive (i.e. logical) level:

- If $\mathit{FrontView} = \mathit{Circle}$, then $\mathit{action} = \mathit{RecognizeSphere}$
- If $\mathit{FrontView} = \mathit{Unknown} \wedge \mathit{SideView} = \mathit{Triangle}$, then $\mathit{action} = \mathit{LookAtFront}$
- If $\mathit{FrontView} = \mathit{Triangle} \wedge \mathit{SideView} = \mathit{Unknown}$, then $\mathit{action} = \mathit{RecognizePrism}$

## 5.2 Recognition of Simulated Real Objects

To analyse Sphinx under more realistic conditions, we set up another environment. We defined shape attributes that are suitable for representing objects within a simple object recognition task and then chose random objects and describe them with these previously defined attributes. These attributes are the input to Sphinx.

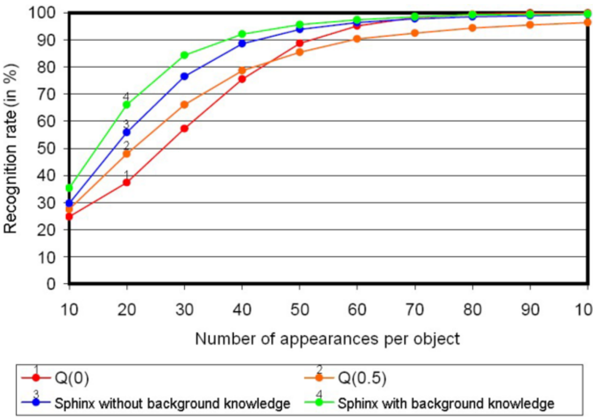

**Figure 2.** Recognition Rates for Geometric Objects

Again, there are three possible perspectives: the front view, the side view, and a view from a position between these two views. The decision for these persepectives, especially for the intermediate view, was made based on the results found by [5] who revealed that the intermediate view plays a special role in human object recognition. The front and the side view are described by three attributes each: approximate (idealized) shape, size (i.e. proportion) of the shape, and deviance from the idealized shape. The approximate shape can take on the values *unknown, circle, square, triangle up*, and *triangle down*. The size can be *unknown, flat, regular*, or *tall*. The deviance can be *little, medium*, or *big*. Besides these attributes the object is described by the complexity of its texture. This attribute can take on the values *simple, medium*, and *complex*. We set the attributes for each object manually. In a real application they can be determined easily by a simple image processing module which merely has to quantize the shape and texture of an object.

If the agent looks at the object from the front or the side, it perceives the matching idealized shape, its size, its deviance, and the complexity of the texture. From the intermediate view the agent can only perceive the idealized shapes of the front and the side view and the complexity of the texture, but not the size and deviances. Formally the domains are:

- $\mathit{FrontViewShape} = \{\mathit{Unknown}, \mathit{Circle}, \mathit{Square}, \mathit{TriangleUp}, \mathit{TriangleDown}\}$
- $\mathit{FrontViewSize} = \{\mathit{Unknown}, \mathit{Flat}, \mathit{Regular}, \mathit{Tall}\}$
- $\mathit{FrontViewDeviation} = \{\mathit{Unknown}, \mathit{Little}, \mathit{Medium}, \mathit{Much}\}$
- $\mathit{SideViewShape} = \{\mathit{Unknown}, \mathit{Circle}, \mathit{Square}, \mathit{TriangleUp}, \mathit{TriangleDown}\}$
- $\mathit{SideViewSize} = \{\mathit{Unknown}, \mathit{Flat}, \mathit{Regular}, \mathit{Tall}\}$
- $\mathit{SideViewDeviation} = \{\mathit{Unknown}, \mathit{Little}, \mathit{Medium}, \mathit{Much}\}$
- $\mathit{Texture} = \{\mathit{Simple}, \mathit{Medium}, \mathit{Complex}\}$
- $\mathit{Action} = \{\mathit{RotateLeft}, \mathit{RotateRight}, \mathit{RecognizeUnkown}\} \cup R$

where $R$ is the set of '*Recognize*' actions. At the beginning of each episode, the agent looks at the current object from a random perspective and the variables are set according to this perspective. Now, the agent can rotate the object clockwise or counter-clockwise or name it. If the agent's action is a '*Recognize*' action, the episode ends. After ten steps the running episode is forced to end. The reward function is the same as in the previous test environment. We have chosen

15 different objects from nine different object classes such as bottle, tree, and house for which we provide the three attributes mentioned (shape, size, and deviation) (see figure 3).

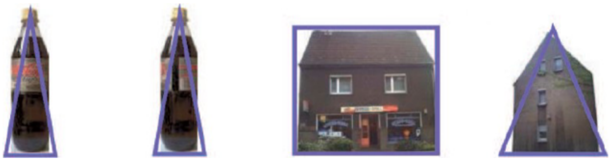

**Figure 3.** Approximated geometrical forms of objects

Similar to the previous scenario, the experimental results obtained by testing 100 independend agents are depicted in Figure 4. Again, it can be seen clearly that SPHINX-Learning does better than $Q(\lambda)$-learning with respect to the speed of learning.

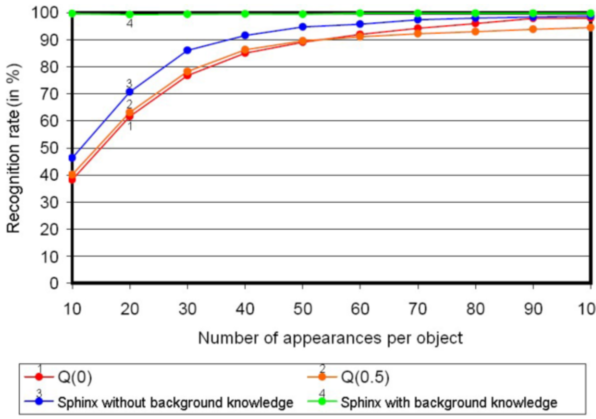

**Figure 4.** Recognition Rates for Simulated Real Objects

In a second step we added background knowledge that enabled the agent to recognize all objects correctly, if it has perceived all of the three views. Furthermore, we added rules to the background knowledge that told the agent to look at the object from all perspectives first. With these rules the agent has a complete, but not optimal, solution for the task. We wanted to find out how fast the agent learns that it does not need all views to classify the current object. To protect the background knowledge from being overwritten by the agent's own rules too early, some parameters were changed, so that the agent had to be more sure about the correctness of a rule before adding it to its belief. This setup resulted in a constantly high recognition rate of over 99 %. The number of perceived views decreased over time from 3.28 to 1.99. The value of 3.28 perceived view vs. 3 possible views results from the fact, that the intermediate view has to be perceived twice if the environment starts in this view. Then, the agent perceives this view at the beginning, then rotates the object to the front and then back to the intermediate view so it can rotate the object to the side view in the next step (or vice versa).

Here are some of the rules the agent learned and assimilated during its training:

- If $\mathit{FrontViewShape} = \mathit{TriangleUp} \wedge \mathit{FrontViewSize} = \mathit{Tall}$, then $\mathit{action} = \mathit{RecognizeBottle}$
- If $\mathit{FrontViewShape} = \mathit{Circle} \wedge \mathit{SideViewShape} = \mathit{Unknown} \wedge \mathit{Texture} = \mathit{Simple}$, then $\mathit{action} = \mathit{RotateLeft}$
- If $\mathit{Texture} = \mathit{Complex}$, then $\mathit{action} = \mathit{RecognizeBottle}$

What remains to be done at this point to apply our system to real images of objects, is the extraction of shape attributes from the images. This can be done by existing segmentation methods.

## 6 CONCLUSION

Both low-level, non-cognitive learning and high-level learning with using epistemic background and acquiring generic knowledge are present in human learning processes. In this paper, we presented the hybrid SPHINX approach that enables intelligent agents to adjust to its environment in a similar way by combining epistemic-based belief revision with experience-based reinforcement learning. We linked both methodologies for two purposes: First, the current epistemic state allows the agent to focus on most plausible actions that are evaluated by QTables to find the most promising actions in some current state. Second, the direct feedback by the environment is used not only to update QTables, but also to generate specific or generic knowledge with which the epistemic state is revised.

In order to illustrate the usefulness of our approach, we described application scenarios from computer vision and performed experiments in which SPHINX agents are employed for object recognition tasks. The evaluation of these experiments shows clearly that the proposed interplay of belief revision and reinforcement learning benefits from the advantages of both methodologies. Therefore, the SPHINX approach allows complex yet flexible interactions between learning and reasoning that help agents perform considerably better.

## REFERENCES

[1] Anderson, J. R., *The architecture of cognition*, Hardvard University Press, Cambridge, MA, 1983.

[2] C. Ye and Yung, N. H. C. and D. Wang, 'A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance', *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **33**(1), 17–27, (2003).

[3] Gombert, J.-E., 'Implicit and explicit learning to read: Implication as for subtypes of dyslexia', *Current Psychology Letters*, **10**(1), (2003).

[4] G. Kern-Isberner, *Conditionals in nonmonotonic reasoning and belief revision*, Springer, LNAI 2087, 2001.

[5] Pereira, A. and James, K. H. and Jones, S. S., and Smith, L. B. Preferred views in children's active exploration of objects, 2006.

[6] Reber, A. S., 'Implicit learning and tacit knowledge', *Journal of Experimental Psychology: General*, **118**(3), 219–235, (1989).

[7] W. Spohn, 'Ordinal conditional functions: a dynamic theory of epistemic states', in *Causation in Decision, Belief Change, and Statistics, II*, eds., W.L. Harper and B. Skyrms, 105–134, Kluwer Academic Publishers, (1988).

[8] Stanley, W. B. and Mathews, R. C. and Buss, R. R. and Kotler-Cope, S., 'Insight without awareness: On the interaction of verbalization, instruction and practice in a simulated process control task', *The Quarterly J. of Exp. Psychology Section A*, **41**(3), 553–577, (1989).

[9] Sun, R. and Merrill, E. and Peterson, T., 'From implicit skills to explicit knowledge: a bottom-up model of skill learning', *Cognitive Science*, **25**(2), 203–244, (2001).

[10] Sun, R. and Slusarz, P. and Terry, C., 'The interaction of the explicit and the implicit in skill learning: A dual-process approach', *Psychological Review*, **112**(1), 159–192, (2005).

[11] Sun, R. and Zhang, X. and Slusarz, P. and Mathews, R., 'The interaction of implicit learning, explicit hypothesis testing learning and implicit-to-explicit knowledge extraction', *Neural Networks*, **20**(1), 34–47, (2007).

[12] Sutton, R. S. and Barto, A. G., *Reinforcement Learning: An Introduction*, Bradford Book, The MIT Press, 1998.