Journal of Machine Learning Research 16 (2015) 1437-1480 Submitted 12/13; Revised 11/14; Published 8/15

A Comprehensive Survey on Safe Reinforcement Learning

Javier García fjgpolo@inf.uc3m.es
Fernando Fernández ffernand@inf.uc3m.es
Universidad Carlos III de Madrid,
Avenida de la Universidad 30,
28911 Leganes, Madrid, Spain

Editor: Joelle Pineau

Abstract

Safe Reinforcement Learning can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes. We categorize and analyze two approaches of Safe Reinforcement Learning. The first is based on the modification of the optimality criterion, the classic discounted finite/infinite horizon, with a safety factor. The second is based on the modification of the exploration process through the incorporation of external knowledge or the guidance of a risk metric. We use the proposed classification to survey the existing literature, as well as suggesting future directions for Safe Reinforcement Learning.

Keywords: reinforcement learning, risk sensitivity, safe exploration, teacher advice

1. Introduction

In reinforcement learning (RL) tasks, the agent perceives the state of the environment, and it acts in order to maximize the long-term return which is based on a real valued reward signal (Sutton and Barto, 1998). However, in some situations in which the safety of the agent is particularly important, for example in expensive robotic platforms, researchers are paying increasing attention not only to the long-term reward maximization, but also to damage avoidance (Mihatsch and Neuneier, 2002; Hans et al., 2008; Martín H. and Lope, 2009; Koppejan and Whiteson, 2011; García and Fernández, 2012).

The safety concept, or its opposite, risk, have taken many forms in the RL literature, and it does not necessarily refer to physical issues. In many works, risk is related to the stochasticity of the environment and with the fact that, in those environments, even an optimal policy (with respect the return) may perform poorly in some cases (Coraluppi and Marcus, 1999; Heger, 1994b). In these approaches, the risk concept is related to the *inherent uncertainty* of the environment (i.e., with its stochastic nature). Since maximizing the long-term reward does not necessarily avoid the rare occurrences of large negative outcomes, we need other criteria to evaluate risk. In this case, the long-term reward maximization is transformed to include some notion of risk related to the variance of the return (Howard and Matheson, 1972; Sato et al., 2002) or its worst-outcome (Heger, 1994b; Borkar, 2002; Gaskett, 2003). In other works, the optimization criterion is transformed to include the probability of visiting *error states* (Geibel and Wysotzki, 2005), or transforming the temporal differences to more heavily weighted events that are unexpectedly bad (Mihatsch and Neuneier, 2002).

Other works do not change the optimization criterion, but the exploration process directly. During the learning process, the agent makes decisions about which action to choose, either to find out more about the environment or to take one step closer towards the goal. In RL, techniques for selecting actions during the learning phase are called exploration/exploitation strategies. Most exploration methods are based on heuristics, rely on statistics collected from sampling the environment, or have a random exploratory component (e.g., $\epsilon - greedy$). Their goal is to explore the state space efficiently. However, most of those exploration methods are blind to the risk of actions. To avoid risky situations, the exploration process is often modified by including prior knowledge of the task. This prior knowledge can be used to provide initial information to the RL algorithm biasing the subsequent exploratory process (Driessens and Džeroski, 2004; Martín H. and Lope, 2009; Koppejan and Whiteson, 2011), to provide a finite set of demonstrations on the task (Abbeel and Ng, 2005; Abbeel et al., 2010), or to provide guidance (Clouse, 1997; García and Fernández, 2012). Approaches based on prior knowledge were not all originally built to handle risky domains but, by the way they were designed, they have been demonstrated to be particularly suitable for this kind of problem. For example, initial knowledge was used to bootstrap an evolutionary approach by the winner of the helicopter control task of the 2009 RL competition (Martín H. and Lope, 2009). In this approach, several neural networks that clone error-free teacher policies are added to the initial population (facilitating the rapid convergence of the algorithm to a near-optimal policy and, indirectly, reducing agent damage or injury). Indeed, as the winner of the helicopter domain is the agent with the highest cumulative reward, the winner must also indirectly reduce helicopter crashes insofar as these incur large catastrophic negative rewards. Although the competition is based on the performance after the learning phase, these methods demonstrate that reducing the number of catastrophic situations, also during the learning phase, can be particularly interesting in real robots where the learning phase is performed in an on-line manner, and not through simulators. Instead, Abbeel and Ng (2005); Abbeel et al. (2010) use a finite set of demonstrations from a teacher to derive a safety policy for the helicopter control task, while minimizing the helicopter crashes. Finally, the guidance provided by a teacher during the exploratory process has also been demonstrated to be an effective method to avoid dangerous or catastrophic states (García and Fernández, 2012). In another line of research, the exploration process is conducted using some form of risk metric based on the *temporal differences* (Gehring and Precup, 2013) or in the weighted sum of an entropy measure and the expected return (Law, 2005).

In this manuscript, we present a comprehensive survey of work which considers the concepts of safety and/or risk within the RL community. We call this subfield within RL, Safe Reinforcement Learning. Safe RL can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes. Safe RL algorithms suffer from the lack of an established taxonomy in which to organize existing approaches. In this survey, we have contributed such a structure, through a categorization of Safe RL algorithms. We segment Safe RL algorithms into two fundamental tendencies. The first consists of transforming the optimization criterion.

The second consists of modifying the exploration process in two ways: (i) through the incorporation of external knowledge, and, (ii) through the use of a risk metric. In this category, we focus on those RL approaches tested in risky domains that reduce or prevent undesirable situations through the modification of the exploration process. The objective of this is to become a starting point for researchers who are initiating their endeavors in Safe RL. It is important to note that the second category includes the first since modifying the optimization criterion will also modify the exploration process. However, in the first category we consider those approaches that transform the optimization criterion in some way to include a form of risk. On the other hand, the optimization criterion in the second category remains, while the exploration process is modified to consider some form of risk.

Resulting from these considerations, the remainder of the paper is organized as follows. Section 2 presents an overview and a categorization of Safe RL algorithms existing in the literature. The methods based on the transformation of the optimization criterion are examined in Section 3. The methods that modify the exploration process by the use of prior knowledge or a risk metric are considered in Section 4. In Section 5 we discuss the surveyed methods and identify open areas of research for future work. Finally, we conclude with Section 6.

2. Overview of Safe Reinforcement Learning

We consider learning in **M**arkov **D**ecision **P**rocesses (MDP) described formally by a tuple $< S, A, T, R >$, where S is the state space, A is the action space, $T : S \times A \rightarrow S$ is the transition function and $R : S \times A \rightarrow \mathbb{R}$ is the reward function (Putterman, 1994). In this survey, we consider two main trends for Safe RL (Table 1) to learn in MDPs. The first one is based on the modification of the optimality criterion to introduce the concept of risk (Section 3). The second is based on the modification of the exploration process to avoid the exploratory actions that can lead the learning system to undesirable or catastrophic situations (Section 4).

Optimization Criterion. As regards the first, the objective of traditional RL algorithms is to find an optimal control policy; that is, to find a function which specifies an action or a strategy for some state of the system to optimize a criterion. This optimization criterion may be to minimize time or any other cost metric, or to maximize rewards, etc. The optimization criterion in RL is described by a variety of terms within the published literature, including the expected return, expected sum of rewards, cumulative reward, cumulative discounted reward or return. Within this article, to avoid terminology misunderstandings, we use the term return.

Definition 1 ***Return***. *The term return is used to refer to the expected cumulative future discounted reward* $R = \sum_{t=0}^{\infty} \gamma^t r_t$*, where* r_t *represents a single real value used to evaluate the selection of an action in a particular state (i.e., the reward), and* $\gamma \in [0, 1]$ *is the discount factor that allows the influence of future rewards to be controlled.*

This optimization criterion is not always the most suitable one in dangerous or risky tasks (Heger, 1994b; Mihatsch and Neuneier, 2002; Geibel and Wysotzki, 2005). There are several alternatives to this optimization criterion in order to consider *risk*. In this survey,

Safe RL	Optimization Criterion	Worst Case Criterion	Inherent Uncertainty		Heger (1994b,a) Gaskett (2003)
			Parameter Uncertainty		Nilim and El Ghaoui (2005) Tamar et al. (2013)
		Risk-Sensitive Criterion	Exponential Functions		Howard and Matheson (1972) Borkar (2001, 2002) Basu et al. (2008)
			Weighted Sum of Return and Risk		Mihatsch and Neuneier (2002) Sato et al. (2002) Geibel and Wysotzki (2005)
		Constrained Criterion			Moldovan and Abbeel (2011, 2012a) Castro et al. (2012) Kadota et al. (2006)
		Other Optimization Criteria			Morimura et al. (2010a,b) Luenberger (2013) Castro et al. (2012)
	Exploration Process	External Knowledge	Providing Initial Knowledge		Driessens and Džeroski (2004) Martín H. and Lope (2009) Song et al. (2012)
			Deriving a Policy from Demonstrations		Abbeel et al. (2010) Tang et al. (2010)
			Teacher Advice	Ask for Help	Clouse (1997) García and Fernández (2012) Geramifard et al. (2013)
				Teacher Provide Advices	Clouse and Utgoff (1992) Thomaz and Breazeal (2006, 2008) Vidal et al. (2013)
				Other Approaches	Rosenstein and Barto (2002, 2004) Kuhlmann et al. (2004) Torrey and Taylor (2012)
		Risk-directed Exploration			Gehring and Precup (2013) Law (2005)

Table 1: Overview of the approaches for Safe Reinforcement Learning considered in this survey.

we categorize these optimization criteria in four groups: (i) the worst-case criterion, (ii) the risk-sensitive criterion, (iii) the constrained criterion, and (iv) other optimization criteria.

- *Worst Case Criterion.* The first criterion is based on the *Worst Case Criterion* where a policy is considered to be optimal if it has the maximum worst-case return (Section 3.1). This criterion is used to mitigate the effects of the variability induced by a given policy, since this variability can lead to risk or undesirable situations. This variability can be due to two types of uncertainties: the *inherent uncertainty* related to the stochastic nature of the system (Heger, 1994b,a; Gaskett, 2003), and the *parameter uncertainty* related to some of the parameters of the MDP are not known exactly (Nilim and El Ghaoui, 2005; Tamar et al., 2013).

- *Risk-Sensitive Criterion.* In other approaches, the optimization criterion is transformed so as to reflect a subjective measure balancing the return and the *risk*. These approaches are known as *risk-sensitive approaches* and are characterized by the presence of a parameter that allows the *sensitivity* to the *risk* to be controlled (Section 3.2). In these cases, the optimization criterion is transformed into an exponential utility function (Howard and Matheson, 1972), or a linear combination of return and *risk*, where *risk* can be defined as the *variance of the return* (Markowitz, 1952; Sato et al., 2002), or as the probability of entering into an *error state* (Geibel and Wysotzki, 2005).

- *Constrained Criterion.* The purpose of this objective is to maximize the return subject to one or more constraints resulting in the *constrained optimization criterion* (Section 3.3). In such a case, we want to maximize the return while keeping other types of expected measures higher (or lower) than some given bounds (Kadota et al., 2006; Moldovan and Abbeel, 2012a).

- *Other Optimization Criteria.* Finally, other approaches are based on the use of optimization criteria falling into the area of financial engineering, such as the *r-squared*, *value-at-risk* (*VaR*) (Mausser and Rosen, 1998; Kashima, 2007; Luenberger, 2013), or the density of the return (Morimura et al., 2010a,b) (Section 3.4).

Exploration Process. As regards the modification of the exploration process, there are also several approaches to overcoming the problems where the exploratory actions may have serious consequences. Most RL algorithms begin learning with no external knowledge of the task. In such cases, exploration strategies such as $\epsilon-greedy$ are used. The application of this strategy results in the random exploration of the state and action spaces to gather knowledge on the task. Only when enough information is discovered from the environment, does the algorithm's behavior improve. The randomized exploration strategies, however, waste a significant amount of time exploring irrelevant regions of the state and action spaces, or lead the agent to undesirable states which may result in damage or injury to the agent, the learning system or external entities. In this survey we consider two ways of modifying the exploration process to avoid risk situations: (i) through the incorporation of external knowledge and, (ii) through the use of a risk-directed exploration.

- *External Knowledge.* We distinguish three ways of incorporating prior knowledge into the exploration process (Section 4.1) by: (i) providing initial knowledge, (ii) deriving a policy from a finite set of demonstrations and, (iii) providing teach advice.

  – *Providing Initial Knowledge.* To mitigate the aforementioned exploration difficulties, examples gathered from a teacher or previous information on the task can be used to *provide initial knowledge* for the learning algorithm (Section 4.1.1). This knowledge can be used to bootstrap the learning algorithm (i.e., a type of initialization procedure). Following this initialization, the system can switch to a Boltzmann or fully greedy exploration based on the values predicted in the initial training phase (Driessens and Džeroski, 2004). In this way, the learning algorithm is exposed to the most relevant regions of the state and action spaces from the earliest steps of the learning process, thereby eliminating the time needed in random exploration for the discovery of these regions.

  – *Deriving a policy from a finite set of demonstrations.* In a similar way, a set of examples provided by a teacher can be used to *derive a policy from demonstrations* (Section 4.1.2). In this case, the examples provided by the random exploration policy are replaced by the examples provided by the teacher. In contrast to the previous category, this external knowledge is not used to bootstrap the learning algorithm, but is used to learn a model from which to derive a policy in an off-line and, hence, safe manner (Abbeel et al., 2010; Tang et al., 2010).

  – *Providing Teach Advice.* Other approaches based on *teacher advice* assist the exploration during the learning process (Section 4.1.3). They assume the availability of a teacher for the learning agent. The teacher may be a human or a simple controller, but in both cases it does not need to be an expert in the task. At every step, the agent observes the state, chooses an action, and receives the reward with the objective of maximizing the return or other optimization criterion. The teacher shares this goal, and provides actions or information to the learner agent. Both the agent and the teacher can initiate this interaction during the learning process. In the *ask for help* approaches (Section 4.1.3.1), the learner agent requests advice from the teacher when it considers it necessary (Clouse, 1997; García and Fernández, 2012). In other words, the teacher only provides advice to the learner agent when it is explicitly asked to. In other approaches (Section 4.1.3.2), it is the teacher who provides actions whenever it feels it is necessary (Thomaz and Breazeal, 2008; Vidal et al., 2013). In another group of approaches (Section 4.1.3.3), the main role in this interaction is not so clear (Rosenstein and Barto, 2004; Torrey and Taylor, 2012).

- *Risk-directed Exploration.* In these approaches a risk measure is used to determine the probability of selecting different actions during the exploration process (Section 4.2) while the classic optimization criterion remains (Gehring and Precup, 2013; Law, 2005).

3. Modifying the Optimization Criterion

This section describes the methods of the first category of the proposed taxonomy based on the transformation of the optimization criterion. The approaches using the return as the objective function are referred to as risk-neutral control (Putterman, 1994), because the variance and higher order moments in the probability distribution of the rewards are neglected.

Definition 2 ***Risk-Neutral Criterion.*** *In risk-neutral control, the objective is to compute (or learn) a control policy that maximizes the expectation of the return,*

$$\max_{\pi \in \Pi} E_\pi(R) = \max_{\pi \in \Pi} E_\pi(\sum_{t=0}^{\infty} \gamma^t r_t), \tag{1}$$

where $E_\pi(\cdot)$ stands for the expectation with respect to the policy π.

The decision maker may be also interested in other objective functions, different from the expectation of the return, to consider the notion of risk. In this case, risk is related to the fact that even an optimal policy may perform poorly in some cases due to the variability of the problem, and the fact that the process behavior is partially known. Because of the latter, the objective function is transformed, resulting in various risk-aware approaches. In this survey, we focus on three optimization criterion: the worst case criterion, the risk-sensitive criterion, and the constrained criterion. These approaches are discussed in detail in the following sections.

3.1 Worst-Case Criterion

In many applications, we would like to use an optimization criterion that incorporates a penalty for the variability induced by a given policy, since this variability can lead to risk or undesirable situations. This variability can be due to two types of uncertainties: a) the *inherent uncertainty* related to the stochastic nature of the system, and b) the *parameter uncertainty*, related to some of the parameters of the MDP are not known exactly. To mitigate this problem, the agent maximizes the return associated to the worst-case scenario, even though the case may be highly unlikely.

3.1.1 Worst-Case Criterion under Inherent Uncertainty

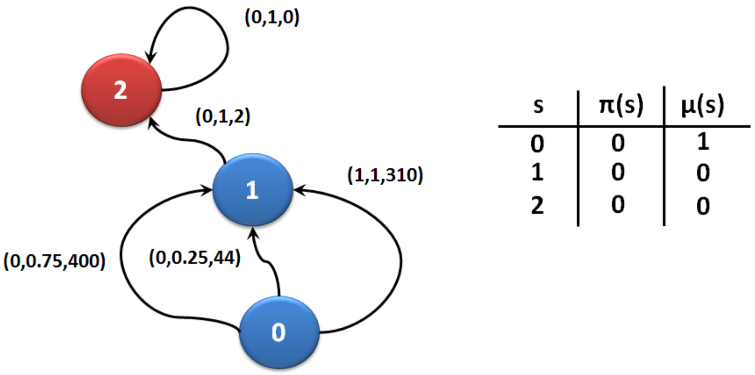
This approach is discussed at length in the literature (Heger, 1994b; Coraluppi, 1997; Coraluppi and Marcus, 1999, 2000).

Definition 3 ***Worst-Case or Minimax Criterion under inherent uncertainty.*** *In worst-case or minimax control the objective is to compute (or learn) a control policy that maximizes the expectation of the return with respect to the worst case scenario (i.e., the worst outcome) incurred in the learning process using,*

$$\max_{\pi \in \Pi} \min_{w \in \Omega^\pi} \; E_{\pi,w}(R) = \max_{\pi \in \Pi} \min_{w \in \Omega^\pi} \; E_{\pi,w}(\sum_{t=0}^{\infty} \gamma^t r_t), \tag{2}$$

where Ω^π is a set of trajectories of the form $(s_0, a_0, s_1, a_1, \dots)$ that occurs under policy π, and where $E_{\pi,w}(\cdot)$ stands for the expectation with respect to the policy π and the trajectory w. That is, we are interested in the policy $\pi \in \Pi$ with the max-min outcome.

We briefly review the difference between the risk-neutral and worst-case criterion using the example provided by Hedger, replicated in Figure 1 (see Heger, 1994b).



s	π(s)	μ(s)
0	0	1
1	0	0
2	0	0

Figure 1: Difference between risk-neutral and worst-case criterion. Example provided by Heger (1994b). Each transition is labeled as a triple. The first number a in the triple is an admissible action for the state s. The second number stands for the probability that the state transition will occur if action a is selected in the corresponding starting state s. The third number represents the immediate reward for the transition.

In Figure 1, there are three states and transitions between them. Each transition is represented by three components: the first is the action that performs the transition from one state to the other, the second is the probability of the transition, and the last is the reward obtained when performing this transition. Additionally, there are two policies labeled π and μ. In Figure 1, $E_\pi(R) = 311 + 2\gamma$ and $E_\mu(R) = 310 + 2\gamma$. Therefore, by applying the risk-neutral criterion, the policy π is optimal. However, $\max \ \inf \ (E_\pi(R)) = 44 + 2\gamma$ and $\max \ \inf \ (E_\mu(R)) = 310 + 2\gamma$. Therefore μ is optimal when applying the worst-case criterion. In worst case control strategies, the optimality criterion is exclusively focused on risk-avoidance or risk-averse policies. A policy is considered to be optimal if its worst-case return is superior.

Heger (1994b) introduces the $\hat{Q} - Learning$ which can be regarded as the counterpart to Q-Learning (Watkins, 1989) related to the minimax criterion,

$$\hat{Q}(s_t, a_t) = \min(\hat{Q}(s_t, a_t), r_{t+1} + \gamma \max_{a_{t+1} \in A} \hat{Q}(s_{t+1}, a_{t+1})) \tag{3}$$

The $\hat{Q}$ value is essentially a lower bound on value. $\hat{Q} - learning$ and the minimax criterion are useful when avoiding risk is imperative. Jiang et al. (1998) combine the simple function approximation state aggregation with the minimax criterion and present the convergence theory for $\hat{Q} - learning$. However, Gaskett (2003) tested $\hat{Q} - learning$ in a stochastic cliff world environment, under the condition that actions are picked greedily, and found that $\hat{Q}-learning$ demonstrated extreme pessimism which can be more injurious than beneficial (Gaskett, 2003). Chakravorty and Hyland (2003) apply the minimax criterion to the actor-critic architecture and presents error bounds when using state aggregation as a function approximation. In general, the minimax criterion is too restrictive as it takes into account severe but extremely rare events which may never occur (Mihatsch and Neuneier, 2002). The $\alpha - value$ of the return $\hat{m_\alpha}$ introduced by Heger (1994a) can be seen as an extension of the worst case control of MDPs. This concept establishes that the returns $R < \hat{m_\alpha}$ of a policy that occur with a probability of less than α are ignored. The algorithm is less pessimistic than the pure worst case control, given that extremely rare scenarios have no effect on the policy.

Gaskett (2003) proposes a new extension to Q-learning, β-pessimistic $Q - learning$, which compromises between the extreme optimism of standard $Q-learning$ and the extreme pessimism of minimax approaches,

$$Q_\beta(s_t, a_t) = Q_\beta(s_t, a_t) + \alpha(r_{t+1} + \gamma((1-\beta)\max_{a_{t+1}\in A} Q_\beta(s_{t+1}, a_{t+1}) + \beta \min_{a_{t+1}\in A} Q_\beta(s_{t+1}, a_{t+1}))) \tag{4}$$

In the β-pessimistic $Q - learning$ algorithm the value of $\beta \in [0, 1]$ renders the equation into the standard $Q - learning$ or the minimax algorithm respectively (Gaskett, 2003). Experimental results show that when $\beta = 0.5$, the algorithm reaches the same level of pessimism as $\hat{Q} - learning$, although the agent manages to reach the goal state in some cases, unlike in $\hat{Q} - learning$.

3.1.2 Worst-Case Criterion under Parameter Uncertainty

Some RL approaches are focused to learning the model first which is assumed to be correct, and then applying a dynamic programming to it to learn and optimal policy. However, in practice, the model learned is typically estimated from noisy data or insufficient training examples, or even worse, they may change during the execution of a policy. These modeling errors may have fatal consequences in real, physical systems, where there are often states that are really catastrophic and must be avoided even during learning. This problem is faced by the robust control community (Zhou et al., 1996), whose one goal is to build policies with satisfactory online performance and robustness to model errors. Specifically, a robust MDP deals with uncertainties in parameters; that is, some of the parameters, namely, transition probabilities, of the MDP are not known exactly (Bagnell et al., 2001; Iyengar, 2004; Nilim and El Ghaoui, 2005).

Definition 4 ***Worst-Case or Minimax Criterion under parameter uncertainty.*** *Typically this criterion is described in terms of a set (uncertainty set), P, of possible transition matrices, and the objective is to maximize the expectation of the return for the worst case policy over all possible models $p \in P$,*

$$\max_{\pi\in\Pi} \min_{p\in P} E_{\pi,p}(R) = \max_{\pi\in\Pi} \min_{p\in P} E_{\pi,p}(\sum_{t=0}^{\infty} \gamma^t r_t), \tag{5}$$

where $E_{\pi,p}(\cdot)$ stands for the expectation with respect to the policy π and the transition model p.

The problem of parameter uncertainty has been recognized in the reinforcement learning community as well, and algorithms have been suggested to deal with it (Tamar et al., 2013). Typical model-based reinforcement algorithms ignore this type of uncertainty. However, minimizing the risk is one of several imports in model-based reinforcement learning solutions, particularly ones in which failure has important consequences (Bagnell and Schneider, 2008), as a learned model invariably has certain inaccuracies, due to insufficient or noise training data (Bagnell, 2004).

3.2 Risk-Sensitive Criterion

In risk-sensitive RL, the agent has to strike a balance between getting large reinforcements and avoiding catastrophic situations even if they occur with very small probability. For example, a profit-maximizing firm may want to be conservative in making business decisions to avoid bankruptcy even if its conservation will probably lower the expected profits.

Definition 5 ***Risk-Sensitive Criterion.*** *In risk-sensitive RL, the objective function includes a scalar parameter β that allows the desired level of risk to be controlled. The parameter β is known as the risk sensitivity parameter, and is generally either positive or negative: $\beta > 0$ implies risk aversion, $\beta < 0$ implies a risk-seeking preference, and (through a limiting argument) $\beta = 0$ implies risk neutrality.*

Depending on the form of the objective function, it is possible to consider various risk-sensitive RL algorithms.

3.2.1 Risk-Sensitive Based on Exponential Functions

In risk-sensitive control based on the use of *exponential utility functions*, the return R is transformed to reflect a subjective measure of utility (Howard and Matheson, 1972; Chung and Sobel, 1987). Instead of maximizing the expected value of R, the objective here is to maximize

$$\max_{\pi \in \Pi} \; \beta^{-1} log E_\pi(\exp^{\beta R}) = \max_{\pi \in \Pi} \; \beta^{-1} log E_\pi(\exp^{\beta \sum_{t=0}^{\infty} \gamma^t r_t}), \tag{6}$$

where β is a parameter and R is the return. A straightforward Taylor expansion of the *exp* and *log* terms of Equation 6 yields in Equation (by using the big $\mathcal{O}$ notation)

$$\max_{\pi \in \Pi} \; \beta^{-1} log E_\pi(\exp^{\beta R}) = \max_{\pi \in \Pi} \; E_\pi(R) + \frac{\beta}{2} Var(R) + \mathcal{O}(\beta^2), \tag{7}$$

where $Var(R)$ denotes the variance of the return. Variability is penalized for $\beta < 0$ and enforced for $\beta > 0$. Therefore, the objective is risk-averse for $\beta < 0$, risk-seeking for $\beta > 0$ and risk-neutral for $\beta = 0$.

Most of the work of this trend is within the MDP framework where the transition probabilities and rewards are explicitly available. As an example, Patek (2001) analyzed a class of terminating MDPs with a risk-averse, expected-exponential criterion, with compact constraint sets. By restricting attention to risk-averse problems ($\beta > 0$) with all transition costs strictly positive, and by assuming the existence of a stationary policy, the authors established the existence of stationary optimal policies. More recently, Osogami (2012) and Moldovan and Abbeel (2012b) demonstrate that a risk-sensitive MDP for maximizing the expected exponential utility is equivalent to a robust MDP for maximizing the worst-case criterion. Although the exponential utility approach constitutes the most popular and best analyzed risk-sensitive control framework in the literature, there remain serious drawbacks which prevent the formulation of corresponding RL algorithms (Mihatsch and Neuneier, 2002): time-dependent optimal policies, and no model-free RL algorithms for both deterministic and stochastic reward structures. As a result, the use of this criterion does not lend itself easily to model-free reinforcement learning methods such as TD(0) or Q-learning (Heger, 1994b). Therefore, much less work has been done within the RL framework using this exponential utility function as an optimization criterion, with a notable exception of Borkar (2001, 2002) who relaxes the assumption of a system model by deriving a variant of the Q-learning algorithm for finite MDPs with an exponential utility. Basu et al. (2008) present an approach, extending the works by Borkar (2001, 2002), for Markov decision processes with an infinite horizon risk-sensitive cost based on an exponential function. Its convergence is proved using the **o**rdinary **d**ifferential **e**quation (o.d.e) method for stochastic approximation, and it is also extended to continuous state space processes.

In a different line of work, Chang-Ming et al. (2007) demonstrated that the max operator in Equation 6 can be replaced with a generalized averaged operator in order to improve the robustness of RL algorithms. From a more practical point of view, Liu et al. (2003) use an exponential function in the context of auction agents. Since companies are often risk-averse, the authors derive a closed form of the optimal bidding function for auction agents that maximize the expected utility of the profit for concave exponential utility functions.

However, all the approaches considered in this trend share the same idea: associate the risk with the variance of the return. Higher variance implies more instability and, hence, more risk. Therefore, it should be noted that the aforementioned approaches are not suited for problems where a policy with a small variance can produce a large risk (Geibel and Wysotzki, 2005).

3.2.2 Risk-Sensitive RL Based on the Weighted Sum of Return and Risk

In this trend, the objective function is expressed as the weighted sum of return and risk given by

$$\max_{\pi \in \Pi} \left(E_{\pi}(R) - \beta\, \omega\right) \tag{8}$$

In Equation 8, $E_{\pi}(R)$ refers to the expectation of the return with respect the policy π, β is the risk-sensitive parameter, and ω refers to the consideration of the risk concept which can take various forms. A general objective function is the well-known Markowitz criterion (Markowitz, 1952) where the ω in Equation 8 is replaced by the variance of the return, $Var(R)$. This criterion is also known in the literature as *variance-penalized criterion* (Gosavi, 2009), *expected value-variance criterion* (Taha, 1992; Heger, 1994b) and *expected-value-minus-variance-criterion* (Geibel and Wysotzki, 2005). Within the RL framework, Sato et al. (2002) propose an approach that directly optimizes an objective function defined as a linear combination of the mean and the variance of the return. However, this is based on the assumption of the mean-variance model where the distribution of the return follows a Gaussian distribution, which does not hold in most situations. There are several limitations when using the return variance as a measure of risk. First, the fat tails of the distribution are not accounted for. Consequently, risk can be underestimated due to the ignorance of low probability, but highly severe events. Second, variance penalizes both positive and negative risk equally and does not distinguish between the two. Third, this criterion is incorrectly applied to many cases in which risk cannot be described by the variance of the return (Szegö, 2005). Additionally, mean minus variance optimization within the MDP framework has been shown to be NP-hard in general, and optimizing this criterion can directly lead to counterintuitive policies (Mannor and Tsitsiklis, 2011).

Mihatsch and Neuneier (2002) replace the ω in Equation 8 with the temporal difference errors that occur during learning. Their learning algorithm has a parameter $\beta \in (-1.0, 1.0)$ that allows for switching between *risk-averse* behavior ($\beta = 1$), *risk-neutral* behavior ($\beta = 0$) and risk-seeking behavior ($\beta = -1$). Loosely speaking, the authors overweigh transitions to successor states where the immediate return happen to be smaller than in the average, and they underweigh transitions to states that promise a higher return than the average. In the study, the authors demonstrate that the learning algorithm has the same limiting behavior as exponential utility functions. This method is extended by Campos to deal with large dimensional state/action spaces (Campos and Langlois, 2003).

Geibel and Wysotzki (2005) replace the ω in Equation 8 with the probability, $\rho^\pi(s)$, in which a state sequence $(s_i)_{i \geq 0}$ with $s_0 = s$, generated by the execution of policy π, terminates in an error state,

$$\rho^\pi(s) = E(\sum_{i=0}^{\infty} \gamma^i \bar{r}) \tag{9}$$

In Equation 9, $\bar{r}$ is a cost function in which $\bar{r} = 1$ if an error state occurs and $\bar{r} = 0$ if not. In this case and, as demonstrated by García and Fernández (2012), $\rho^\pi(s)$ is learned by TD methods which require error states (i.e., helicopter crashes or company bankruptcies) to be visited repeatedly in order to approximate the risk function and, subsequently, to avoid dangerous situations.

Common to the works of Mihastch and Geibel is the fact that risk-sensitive behavior is induced by transforming the action values, $Q(s, a)$, or the state values, $V(s)$. There are several reasons why this may not be desirable: (i) if these values are updated based on a conservative criterion, the policy may be overly pessimistic; (ii) the worst thing that can happen to an agent in an environment may have high utility in the long term, but fatal consequences in the short term; and (iii) the distortion of these values means that the true long term utility of the actions are lost.

3.3 Constrained Criterion

The constrained criterion is applied in the literature to *constrained* Markov processes in which we want to maximize the expectation of the return while keeping other types of expected utilities lower than some given bounds (Altman, 1992). This approach might be considered within the second category of the taxonomy described here, since the optimization criterion remains. However, the addition of constraints to this optimization criterion is sufficient to consider a transformation and so we consider that it must be included within this category. The *constrained* MDP is an extension of the MDP framework described as the tuple $< S, A, R, T, C >$, where S, A, R, T are defined as in standard MDP, and C is a set of constraints applied to the policy.

Definition 6 ***Constrained Criterion.*** *In the constrained criterion, the expectation of the return is maximized subject to one or more constraints, $c_i \in C$. The general form of this criterion is shown in the following*

$$\max_{\pi \in \Pi} E_\pi(R) \text{ subject to } c_i \in C, c_i = \{h_i \leq \alpha_i\}, \tag{10}$$

where c_i represents the ith constraint in C that the policy π must fulfill, with $c_i = \{h_i \leq \alpha_i\}$ where h_i is a function related with the return and α_i is the threshold restricting the values of this function. Depending of the problem the symbol $\leq$ in the constraints $c_i \in C$ may be replaced by $\geq$.

We can see the proposed constraints in Equation 10 as restrictions on the space of allowable policies. Figure 2 shows the entire policy space, Π, and the set of allowable policies, $\Gamma \subset \Pi$, where each policy $\pi \in \Gamma$ satisfies the constraints $c_i \in C$.

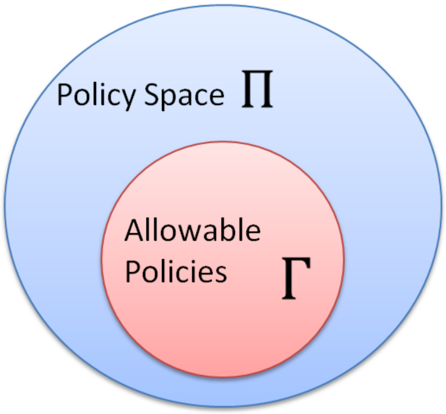


Figure 2: Policy space, Π, and the set of allowable policies, $\Gamma \subset \Pi$, where each policy $\pi \in \Gamma$ satisfies the constraints $c_i \in C$.

Therefore, Equation 10 can be transformed into

$$\max_{\pi \in \Gamma} E_\pi(R) \tag{11}$$

From a safety point of view, this optimization criterion is particularly suitable for risky domains. In these domains, the objective may be seen as finding the best policy π in the space of considered safe policies, Γ. This space, Γ, may be restricted using various types of constraints: constraints to ensure that the expectation of the return exceeds some specific minimum threshold, to ensure that the variance of the return does not exceed specific maximum threshold, to enforce *ergodicity*, to ensure specific restrictions of the problem.

A typical constraint is referred to ensure that the expectation of the return exceeds some specific minimum threshold, $E(R) \geq \alpha$ (Geibel, 2006). In this case, the space of considered safe policies, Γ, is made up of the policies for which the expectation of return exceeds the specific threshold, α. This is suitable in situations in which we already know a reasonably good policy, and we want to improve it through exploration, but the expectation of the return may not fall below a given safety margin. In these kinds of problem, one can derive the LP problem by using a Lagrangian approach which allows us to transform the constrained problem into a equivalent non-constrained one. As an example, Kadota et al. (2006) transform the constrained criterion into a Lagrangian expression. In this way, the method reduces the constrained problem with n variables to one with $n + k$ unrestricted variables, where k is equal to the number of restrictions. Thus, the resulting expression can be solved more easily. The previous constraint is a hard constraint that cannot be violated, but other approaches allow a certain admissible chance of constraint violation. This chance-constraint metric, $P(E(R) \geq \alpha) \geq (1 - \epsilon)$, is interpreted as guaranteeing that the expectation of the return (considered a random variable) will be at least as good as α with a probability greater than or equal to $(1 - \epsilon)$ (Delage and Mannor, 2010; Ponda et al., 2013).

Instead, other approaches use a different constrained criterion in which the variance of the return must not exceed a given threshold, $Var(R) \leq \alpha$ (Castro et al., 2012). In this case, the space of safe policies, Γ, is made up of policies for which the variance does not exceed a threshold, α. This constrained problem is also transformed into an equivalent unconstrained problem by using *penalty methods* (Smith et al., 1997). Then, the problem is solved using standard unconstrained optimization techniques.

Other approaches rely on *ergodic* MDPs (Hutter, 2002) which guarantee that any state is reachable from any other state by following a suitable policy. Unfortunately, many risky domains are not *ergodic*. For example, our robot helicopter learning to fly cannot recover on its own after crashing. The space of safe policies, Γ, is restricted to those policies that preserve ergodicity with some user-specified probability, α, called the safety level. That is, only visiting states s so that one can always get back from s to the initial state (Moldovan and Abbeel, 2011, 2012a). In this case, the authors use plain linear programming formulation after removing the non-linear dependences to solve the constrained MDP efficiently. It is important to note that this constrained criterion is closely related to the *recoverable* and *value-state* concepts described by Ryabko and Hutter. An environment is *recoverable* if it is able to forgive initial *wrong* actions, i.e., after any arbitrary finite sequence of actions, the optimal policy is still achievable. Additionally, an environment is *value-stable* if from any sequence of k actions, it is possible to return to the optimal level of reward in $o(k)$ steps; that is, it is not just possible to recover after any sequence of (wrong) actions, but it is possible to recover fast.

Finally, Abe et al. (2010) proposed a constrained RL algorithm and reported their experience in an actual deployment of a tax collection optimization system based on their approach, at New York State Department of Taxation and Finance. In this case, the set of constraints, C, is made up of legal, business and resource constraints, which are specific to the problem under consideration. In contrast to the previous general formulations in which the constraints are defined as a function of the entire state trajectory, the authors formulate the constraints as being fixed and known at each learning iteration.

However these approaches have three main drawbacks. First, the correct selection of the threshold α. Higher values mean that they are too permissive, or conversely, too restrictive. Second, they do not prevent the fatal consequences in the short term. Finally, these methods associate the risk to policies in which the return or its variance is greater than a specified threshold, which is not suitable for most risk domains.

3.4 Other Optimization Criteria

In the area of financial engineering, various risk metrics such as *r-squared*, *beta*, *Sharpe ratio* or *value-at-risk* (*VaR*) have been studied for decision making with a low risk of huge costs (Mausser and Rosen, 1998; Kashima, 2007; Luenberger, 2013). Castro et al. (2012) also use the *Sharpe ratio* criterion, $\max_{\pi\in\Pi} E_\pi(R)\sqrt{Var(R)}$. The performance of this criterion is compared with the constrained criterion, $Var(R) \leq \alpha$, and the classic optimization criterion (Equation 1) in a portfolio management problem where the available investment options include both liquid and non-liquid assets. The non-liquid asset has some risk of not being paid (i.e., a default) with a given probability. The policy for the classic criterion is risky, and yields a higher gain than the policy for the constrained criterion. Interestingly, $\max_{\pi\in\Pi} E_\pi(R)\sqrt{Var(R)}$ resulted in a very risk-averse policy, that almost never invested in the non-liquid asset. This interesting phenomenon discourages the use of this optimization criterion. Even the authors suggest that it might be more prudent to consider other risk measures instead of the $\max_{\pi\in\Pi} E_\pi(R)\sqrt{Var(R)}$. Morimura et al. (2010a,b) focus their risk-sensitive approach on estimating the density of the returns, which allows them to handle various risk-sensitive criteria. However, the resulting distributional-SARSA-with-CVaR (or d-SARSA with CVaR) algorithm, has proved effectiveness only in a very simple and discrete MDP with 14 states.

4. Modifying the Exploration Process

This section describes the methods of the second category of the proposed taxonomy. In this category, in contrast with the previous one, the optimization criterion remains, but the exploration process is modified to consider some form of risk. Classic exploration/exploitation strategies in RL assume that the agent must explore and learn everything from scratch. In this framework, the agent is blind to the risk of actions during learning, potentially ending up in catastrophic states (Geibel and Wysotzki, 2005; García and Fernández, 2012). The helicopter hovering control task is one such case involving high risk, since some policies can crash the helicopter, incurring catastrophic negative reward. Exploration/exploitation strategies such as $\epsilon - greedy$ may even result in constant helicopter crashes (especially where there is a high probability of random action selection). In addition, random exploration policies waste a significant amount of time exploring irrelevant regions of the state and action spaces in which the optimal policy will never be encountered. This problem is more pronounced in environments with extremely large and continuous state and action spaces. Finally, it is impossible to completely avoid undesirable situations in high-risk environments without a certain amount of external knowledge (that is, not coming from interaction between the agent and the system): the use of random exploration would require an undesirable state to be visited before it can be labeled as undesirable. However, such visits to undesirable states usually lead to unrecoverable situations or *traps* (Ryabko and

Hutter) (i.e., the agent is not able to achieve the optimal policy after a sequence of *wrong* actions) and may result in damage or injury to the agent, the learning system or external entities. Consequently, visits to these states should be avoided from the earliest steps of the learning process. In this paper, we focus on two ways of modifying the exploration process in order to avoid visits to undesirable states: through the incorporation of external knowledge or through a directed exploration based on a risk measure. Both approaches are discussed in detail in the following sections.

4.1 Incorporating External Knowledge

Mitigating the difficulties described above, external knowledge (e.g., finite sets of teacher-provided examples or demonstrations) can be used in three general ways, either (i) to provide initial knowledge (i.e., a type of initialization procedure) or (ii) to derive a policy from a finite set of examples or (iii) to guide the exploration process through teacher advice. In the first case, the knowledge is used to bootstrap the value function approximation and lead the agent through the more relevant regions of the space. In the second way, finite sets of teacher-provided examples or demonstrations can be used to derive a policy. In these ways, the learning algorithm is exposed to the most relevant regions of the state and action spaces from the earliest steps of the learning process, thereby eliminating the time needed in random exploration for the discovery of these regions.

However, while furnishing the agent with initial knowledge helps to mitigate the problems associated with random exploration, this initialization alone is not sufficient to prevent the undesirable situations that arise in the subsequent explorations undertaken to improve learner ability. An additional mechanism is necessary to guide this subsequent exploration process in such a way that the agent may be kept far away from catastrophic states. So, in the third case, a teacher is used to provide information when it is considered necessary. These three ways of incorporating external knowledge are widely discussed in the following sections. Some of the approaches described here were not created originally as specific Safe RL methods, but they have some properties that make them particularly suitable for these kinds of problem.

4.1.1 Providing Initial Knowledge

The most elementary method for biasing learning is to choose some initialization based on prior knowledge of the problem. In Driessens and Džeroski (2004), a bootstrapping procedure is used for relational RL in which a finite set of demonstrations are recorded from a human teacher and later presented to a regression algorithm (Driessens and Džeroski, 2004). This allows the regression algorithm to build a partial Q-function which can later be used to guide further exploration of the state space using a Boltzmann exploration strategy. Smart and Kaelbling also use examples, training runs to bootstrap the Q-learning approach for their HEDGER algorithm (Smart and Kaelbling, 2000). The initial knowledge bootstrapped into the Q-learning approach allows the agent to learn more effectively and helps to reduce the time spent with random actions. Teacher behaviors are also used as a form of *population seeding* in neuroevolution approaches (Siebel and Sommer, 2007). Evolutionary methods are used to optimize the weights of neural networks, but starting from a prototype network whose weights correspond to a teacher (or baseline policy). Using this technique, RL Competition helicopter hovering task winners Martín H. and Lope (2009) developed an evolutionary RL algorithm in which several teachers are provided in the initial population. The algorithm restricts crossover and mutation operators, allowing only slight changes to the policies given by the teachers. Consequently, it facilitates a rapid convergence of the algorithm to a near-optimal policy, as is the indirect minimization of damage to the agent. In Koppejan and Whiteson (2009, 2011), neural networks are also evolved, beginning with one whose weights correspond to the behavior of the teacher. While this approach has been proven advantageous in numerous applications of evolutionary methods (Hernández-Díaz et al., 2008; Koppejan and Whiteson, 2009), Koppejan's algorithm nevertheless seems somewhat ad-hoc and designed for a specialized set of environments.

Maire (2005) propose an approach for deriving high quality initial value functions from existing demonstrations by a teacher. The resulting value function constitutes a starting point for any value function-based RL method. As the initial value function is substantially more informative than a random value function initialization frequently used with RL methods, the remaining on-line learning process is conducted safer and faster. Song et al. (2012) also improve the performance of the Q-learning algorithm initializing the Q-values appropriately. These approaches are used in the Grid-World domain and are able to reduce drastically the times the agent moves into an obstacle.

Some *Transfer Learning* (TL) algorithms are also used to initialize a learner in a target task (Taylor and Stone, 2009). The core idea of transfer is that experience gained in learning to perform one task can help to improve learning performance in a related, but different, task. Taylor and Stone (2007) train an agent in a source task recording the agent's trajectories (i.e., state-action pairs). Then, the agent uses this experience to train in the target task off-line before the on-line training begins. These authors also learn an action-value function in a source task, translate the function into a target task via a hand-coded inter-task mapping, and then use the transferred function to initialize the target task agent (Taylor et al., 2007). Despite TL approaches having been shown effective in speeding up the learning processes, they present two main difficulties in their applicability to risky domains: (i) the knowledge to be reused in the target task requires it to be previously learned in a source task(s) (which is not always possible to do in a safe manner), and (ii) it is not always trivial to transfer this knowledge from the source task(s) to the target task since they could be of a different nature.

There is extensive literature on initialization in RL algorithms (Burkov and Chaib-draa, 2007), but their intensive analysis falls outside the scope of this paper since not all of them are focused to preserving the agent's safety or avoiding risky or undesirable situations. But the bias introduced in the learning process and the rapid convergence produced by most of these algorithms, can ensure their applicability to risky domains. However, this approach is problematic for two main reasons. First, if the initialization does not provide information for all important states the agent may end up with a suboptimal policy. Second, the exploration process following the initial training phase can result in visiting new states for which the agent has no information on how to act. As a result, the probability of incurring damage or injury is greatly increased. In addition, the relevance of these methods is highly dependent on the internal representations used by the agent. If the agent simply maintains a table, initialization is easy, but if the agent uses a more complex representation, it maybe very difficult or impossible to initialize the learning algorithm.

4.1.2 Deriving a Policy from a Finite Set of Demonstrations

All approaches falling under this category are framed according to the field of Learning from Demonstration (LfD) (Argall et al., 2009). Highlighting the study by Abbeel and Ng (2005); Abbeel et al. (2010) based on apprenticeship learning, the approach is made up of three distinct steps. In the first, a teacher demonstrates the task to be learned and the state-action trajectories of the teacher's demonstration are recorded. In the second step, all state-action trajectories seen so far are used to learn a model from the system's dynamics. For this model, a (near-)optimal policy is to be found using any reinforcement learning (RL) algorithm. Finally, the policy obtained should be tested by running it on the real system. In Tang et al. (2010), an algorithm based on apprenticeship learning is also presented for automatically-generating trajectories for difficult control tasks. The proposal is based on the learning of parameterized versions of desired maneuvers from multiple expert demonstrations. In these approaches, the learner is able to exceed the performance of the teacher. Despite each approach's potential strengths and general interest, all are inherently linked to the information provided in the demonstration data set. As a result, learner performance is heavily limited by the quality of the teacher's demonstrations. While one way to circumvent the difficulty and improve performance is by exploring beyond what is provided in the teacher demonstrations, this again raises the question of how the agent should act when it encounters a state for which no demonstration exists. One possible answer to this question is based on the use of teacher advice techniques, as defined below.

4.1.3 Using Teacher Advice

Exploring the environment while avoiding fatal states is critical for learning in domains where a bad decision can lead the agent to a dangerous situation. In such domains, different ways of teacher advice in reinforcement learning has been proposed as a form of safe exploration (Clouse, 1997; Hans et al., 2008; Geramifard et al., 2013; García and Fernández, 2012). The guidance provided by a teacher supports the safe exploration in two ways. First, the teacher can guide the learner in promising parts of the state space where suggested by the teacher's policy. This guidance reduces the sample complexity of learning techniques which is important when dealing with dangerous or high-risk domains. Secondly, the teacher is able to provide advice (e.g., safe actions) to the learner when either the learner or the teacher considers it is necessary so as to prevent catastrophic situations.

The idea of a program learning from external advice was first proposed in 1959 by John McCarthy (Mccarthy, 1959). Teacher advice is based on the use of two fundamental sources of training information: future payoff achieved by taking actions according to a given policy from a given state (derived from classic exploration in RL), and the advice from a teacher as regards which action to take next (Utgoff and Clouse, 1991). The objective of the approaches considered here is to combine these two sources of training information. In these approaches, a learner agent improves its policy based on the information (i.e., the advice) provided by a teacher.

Definition 7 ***Teacher Advising (VN and Ravindran, 2011).*** *Any external entity which is able to provide an input to the control algorithm that could be used by the agent to take decisions and modify the progress of its exploration.*

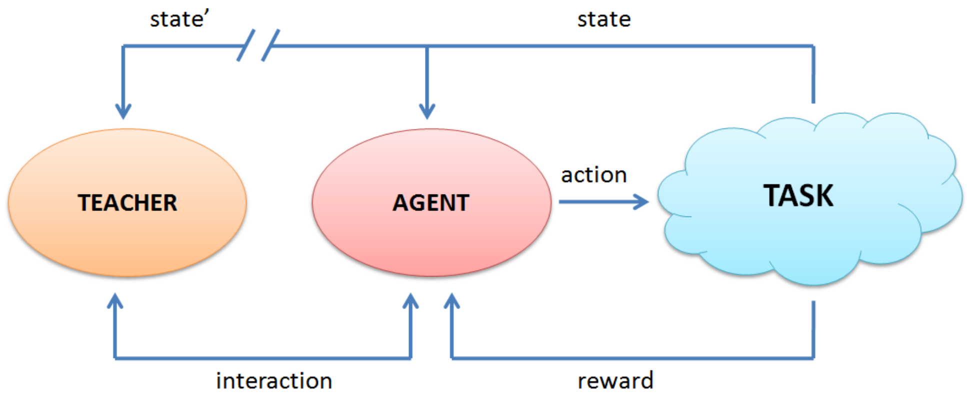


Figure 3: General Teacher-Learner Agent Interaction Scheme.

Figure 3 details such an interaction scheme between the teacher and the learner agent. In every time step, the learner agent perceives the state, chooses the action to perform, and receives a reward as in the classic RL interaction. In this framework, the teacher generally observes the same state as the learner and either the learner or the teacher determines when it is appropriate for the teacher to give an advice. However, the state observed by the agent, *state*, and the teacher, $state'$, could be different if they have different sensing mechanisms (e.g., a robot learner's camera will not detect state changes in the same way as a human teacher's eyes) (Argall et al., 2009). Additionally, the nature of the advice can have various forms: a single action that the learner carries out at that time (Clouse and Utgoff, 1992; Clouse, 1997; García and Fernández, 2012); a complete sequence of actions that the learner agent replays internally (Lin, 1992; Driessens and Džeroski, 2004); reward used to judge the agent's behavior interactively (Thomaz and Breazeal, 2006; Knox and Stone, 2009, 2010; Knox et al., 2011); a set of actions from which the agent has to select one randomly or greedily (Thomaz and Breazeal, 2006; Cetina, 2008).

The general framework of teacher's advice includes five main steps (Philip Klahr Hayes-Roth and Mostow., 1981): (i) requesting or receiving the advise; (ii) converting advice into a usable form; (iii) integrating the reformulated advice into the agent's knowledge base; and (iv) judging the value of advice. In this survey, we focus on step one to classify the different approaches of this trend. Thus, there are two main categories of algorithms: the learner agent asks for advice from the teacher when it needs to, the teacher provides advice to the learner agent when it is necessary.

4.1.3.1 The Learner Agent Asks for Advice

In this approach, the learner agent poses a *confidence parameter* and when this confidence in a state is low, the learner agent asks for advice from the teacher. Typically, this advice corresponds to the action that the teacher would carry out if it were in the place of the learner agent in the current state. In case of advice, the learner agent assimilates the teacher's action by first performing the action (as the learner itself selected it), and later receiving the corresponding reward. This reward is used to update the policy using any RL algorithm. These teacher-advising algorithms are called *Ask for help* algorithms (Clouse, 1997). In the original *Ask for help* approach, Clouse (1997) uses the *confidence parameter* in two different strategies: uniform asking strategy and uncertainty asking strategy. In the first, the learner's request is spread uniformly throughout the learning process. In this case, this parameter establishes the percentage of time steps in which the learner requests help. The second is based on the learner agent's uncertainty about its current action selection. In this case, Clouse establishes that the agent is unsure about the action to choose when all the actions in the current learning step have similar *Q-values*; i.e., if the minimum and maximum *Q-values* are very similar (which is specified by the *confidence parameter*), the learner agent asks for advice. However, this interval-estimation measure between the highest and lowest Q-values produces counterintuitive results in some domains, such the maze domain. In this domain, the true Q-values of actions for each state are very similar since the maze states are highly connected. Interval estimation is therefore not a stable measure of confidence for maze-like domains.

Hans et al. (2008) and García and Fernández (2011, 2012); García et al. (2013) use this *confidence parameter* to detect *risky* situations. In this case, the concept of risk is based on the definition of *fatal transitions* or *unknown states.* Hans et al. (2008) consider a transition to be fatal if the corresponding reward is less than a given threshold τ, and an action a to be unsafe in a given state s if it leads to a fatal transition. In this work, the authors also build the teacher's policy with an altered Bellman optimality equation that does not maximize the return, but the minimal reward to come. The learner agent tries to explore all actions considered safe for all states using the teacher policy or previously identified safe actions in a level-based exploration strategy, which requires storing large amounts of tuples.

García and Fernández (2011, 2012); García et al. (2013) present a new definition of risk based on *unknown* and *known* space, and that reflects the author's intuition as to when human learners require advice. Certainly, humans benefit from help when they are in novel or unknown situations. The authors use a case-based approach to detect such situations. Traditionally, case-based approaches use a *density threshold* θ in order to determine when a new case should be added to the memory. When the distance of the nearest neighbor to the query state is greater than θ, a new case is added. García and Fernández (2011, 2012) propose the PI-SRL algorithm in which a risk function, $\varrho^B(s)$, measures the risk of a state in terms of its similarity to previously visited (and secure) states in a case base, $B = \{c_1 \ldots, c_n\}$. Every case c_i consists of a state-action pair (s_i, a_i) the agent has experienced in the past and with an associated value $V(s_i)$. When the distance to the closest state in the case base is larger than a parameter θ, the risk is maximum, while the risk is minimal if this distance is less than θ. Therefore, in that work, the risk function is defined as a step function. However, to define the risk function in such a way demonstrates that it may still produce damage in the learning agent. The reason is that to follow the teacher's advice only when the distance to the closest known state is larger than θ may be too late. On the other hand, one would expect that the risk function is progressive. Therefore, while the limit of θ is approaching, the risk should start to grow, and the learning agent could start to use the teacher's advice. In a further work, García et al. (2013) propose the use of a progressive risk function that determines the probability of following the teacher advice. The integration of this advice together with the π-reuse exploration strategy (Fernández and Veloso, 2006; Fernández et al., 2010) results in the PR-SRL algorithm (García et al., 2013). The π-reuse exploration strategy allows the agent to use a past policy, Π_{past}, with probability ϕ, explores with probability ϵ, and exploits the current policy, Π_{new}, with probability $1 - \phi - \epsilon$. In the PR-SRL algorithm the past policy Π_{past} is replaced by the teacher policy, the new policy to be learned Π_{new} is replaced by the case base policy in B, and the parameter ψ is replaced by a sigmoid risk function, $\varrho^B(s)$, which computes the probability of teacher's advice.

Hailu and Sommer (1998) also associates the concept of risk to the concept of distance to distinguish novel situations. In this case, the learner agent consists of a feedforward neural network made up of RBF neurons in the input layer and a stochastic neuron in the output layer. Each neuron represents a localized receptive field of width $\sum$ that covers a hyper-sphere of the input space. The learner agent has no neurons at the beginning of the learning process. At this point, the robot perceives a new state s, and it cannot generalize the situation. Therefore, it invokes the teacher which sends its action to the learner. The learner receives the action and adds a neuron. When a new state is perceived, the learner identifies the first winning neuron closest to the state perceived. If the distance of the winning neuron is larger than $\sum$, the state is regarded as novel and the learner invokes the teacher and adds a neuron that generalized the new situation perceived. In this way, the learner grows gradually, thus increasing its competence.

Instead, Geramifard et al. (2011); Geramifard (2012); Geramifard et al. (2013) assume the presence of a function named safe: $S \times A \rightarrow \{0, 1\}$ that returns *true* if the carrying out of action a at state s will result in a catastrophic outcome and *false* otherwise. At every time step, if the learner agent considers the action to be safe, it will be carried out during the next step, otherwise the learner invokes the teacher's action which is assumed to be safe. The safe function is based on the existence of a *constrained* function: $S \rightarrow \{0, 1\}$, which indicates whether being in a particular state is allowed or not. Risk is defined as the probability of visiting any of the constrained states. However, this approach presents two main drawbacks: (i) modeling the *constrained* function correctly, and (ii) it assumes the system model is known or partially known although it is only used for risk analysis. Jessica Vleugel and Gelens (2011) proposed an approach where the unsafe states are previously labeled. In this method, the learner agent asks for advice when it reaches a previously labeled unsafe state.

Finally, although more related to learning from demonstration, Chernova and Veloso (2009) also use the confidence parameter to select between agent autonomy and a request for a demonstration based on the measure of action-selection confidence returned by a classifier. Confidence below a given threshold indicates that the agent is uncertain about which action to take, so it seeks help from the teacher in the form of a demonstration, improving the policy and increasing the confidences for future similar states.

4.1.3.2 The Teacher Provides Advice

In the approaches grouped in this trend, the teacher provides actions or information whenever the teacher feels its help is necessary. Therefore in all these approaches, an explicit mechanism in the learner agent to recognize (and express) its need for advice is not necessary. Therefore, a new open question arises namely what is the best time for a teacher to provide information. Clouse and Utgoff (1992) add a simple interface to a RL algorithm to allow a human teacher to interact with the learner agent during the learning process. In this work, the teacher monitors the learner's behavior and provides an action when it considers it necessary. This action is supposed to be the correct choice to be made in that state. Otherwise, the learner agent takes its own action based on its developing policy. Maclin and Shavlik (1996) use a similar approach in their RATLE algorithm, where a teacher at any point can interrupt the agent learning execution and types its advice using simple IF-THEN rules and more complex rules involving multiple steps. Thomaz and Breazeal (2006, 2008) also introduce an interface between the human teacher and the learner agent. The human teacher advises the agent in two ways: using an interactive reward interface and sending human advice messages. Through the first, the teacher introduces a reward signal $r \in [-1, 1]$ for each step of the learning process. The human teacher receives visual feedback enabling him/her to tune the reward signal before sending it to the agent. Through the second, the agent begins each iteration of the learning loop by pausing to allow the teacher time to introduce advice messages (1.5 seconds). If advice messages are received, the agent will choose randomly between the set of actions derived from these messages. Otherwise, the agent chooses randomly between the set of actions with the highest *Q-values*. In a similar way, Suay and Chernova (2011) also use a teacher to provide rewards and guidance to an Aldebaran Nao humanoid robot. Instead, Vidal et al. (2013) presents a learning algorithm in which the reinforcement comes from a human teacher that is seeing what the robot does. This teacher is able to punish the robot by simply pressing a button on a wireless joystick. When the teacher presses the button to give the robot negative reinforcement, the robot learns from it and transfers the control to the teacher, so that the teacher will be able to move the robot and place it in a suitable position to go on learning. Once this manual control is over, the teacher will press a second button to continue the learning process. In all these approaches, the teacher decides when to provide information based on him/her own feelings (i.e., when the teacher deems it necessary), but there is no metric or rule as to the best time for to do it. Additionally, using the constant monitoring of the learner agent by the teacher, it might not be desirable in practice due to the time or cost implications.

Maclin et al. (2005b) implement the advice as a set of rules provided by a teacher. When a rule applies (i.e., the LHS is satisfied) it is used to say the *Q-value* for some action should be *high* or *low*. The experiments are conducted using the keep away domain, and an example of the rule suggests the keeper with the ball should hold it when the nearest taker is at least 8 metres away (i.e., $Q(hold) \geq high$). In a later work, Maclin et al. (2005a) extends the previous approach to recommend that some action is preferred over another in a specified set of states. Therefore, the teacher is giving advice on policies rather than *Q-values*, which is a more natural way for humans to present advice (e.g., when the nearest taker is at least 8 meters, holding the ball is preferred to passing it). Similarly, Torrey et al. (2005) also used a set of rules, but these rules were learned in a previous related task using inductive logic programming following a transfer learning approach (Taylor and Stone, 2009). The user can also add supplementary teacher advice on the learned rules before the learning process begins. During the learning process, the learner agent receives the teacher's advice and the agent can follow it, refine it, or ignore it according to its value.

Walsh et al. (2011) use a teacher that analyzes the return of the learner agent for each episode. This return provides enough information for the teacher to decide whether or not to provide a demonstration. For each episode, if the return of the agent is lower than a certain measurement, the teacher decides to show a demonstration of that episode starting at the same initial state. In this way, the agent learns concepts that it cannot efficiently learn on its own.

4.1.3.3 Other Approaches

In other approaches, the control of the teacher and the learner agent in the advice-taking interaction is not pre-defined. Rosenstein and Barto (2002, 2004) present a supervised RL algorithm that computes a composite action that is a weighted average of the action suggested by the teacher (a_T) and the exploratory action suggested by the evaluation function (a_E) using

$$a = ka_E + (1-k)a_T \tag{12}$$

In Equation 12, $a_E = a_A + N(0, \sigma)$, where a_A is the action derived from the policy of the agent, $\pi_A(s)$, for a given state s; and $N(0, \sigma)$ is a normal distribution. The parameter k can be used to interpolate between an *ask for help* approach and an approach in which the teacher has the main role. Therefore, k determines the level of control, or autonomy, on the parts of the learner agent and the teacher. On the one hand, the learner agent can set $k = 1$ if it is fully confidence about the action to be taken. Instead, it can set the value of k close to 0 whenever it needs help from its teacher obtaining an *ask for help* approach (Section 4.1.3.1). On the other hand, the teacher can set $k = 0$ whenever it loses confidence in the autonomous behavior of the learner agent similarly to the approaches in Section 4.1.3.2. It is important to note that the proposed way of combining the action suggested by the teacher and the exploratory action is originally conceived for continuous action spaces but it could be extended to a discrete action space as well, by considering distributions over actions.

Kuhlmann et al. (2004) also computes an action using the suggestions of the teacher and the agent. In this work, the teacher generates values for the possible actions in the current world state. It is implemented as a set of rules. If a rule applies, the corresponding action value is increased or decreased by a constant amount. The values generated by the teacher are added to those generated by the learning agent. The final action selected is the action with the greatest final composite value. On the other hand, Judah et al. (2010) use a teacher that is allowed to observe the execution of the agent's current policy and then scroll back and forth along the trajectory and mark any of the available actions in any state as good or bad. The learner agent uses these suggestions and the trajectories generated by itself to compose the agent policy that maximizes the return in the environment. Moreno et al. (2004), by extending the approach proposed by Iglesias et al. (1998b,a), computes an action as the combination of the actions suggested by different teachers. At every time step, each teacher produces a vector of utilities u which contains a value $u(s, a_i) \in [0, 1]$ for each action a_i that it is possible to carry out in the current state s. Then the teacher's vectors are amalgamated into a one single vector, $w(a)$. This vector and an exploratory vector, $e(a)$, computed as $e(a) = 1 - w(a)$, are used to draw up a final *decision vector* which indicates which actions are the most suitable for the current state.

Torrey and Taylor (2012) present an algorithm in which the advice probability depends on the relationship between the learner agent's confidence and the teacher's confidence. In states where the teacher has much greater confidence than that of the student, it gives advice with a greater probability. As the agent's confidence in a state grows, the advice probability decreases. Other approaches are based on interleaving episodes carried out by a teacher with normal exploration episodes. This mixture of teacher and normal exploration makes it easier for the RL algorithms to distinguish between beneficial and poor or unsafe actions. Lin (1991, 1992) use two different interleaving strategies. In the first, after each complete episode, the learner agent replays n demonstrations chosen randomly from the most recent 100 experienced demonstrations, with recent lessons exponentially more likely to be chosen. In the second, after each complete episode, the agent also stochastically chooses already taught demonstrations for replay. Driessens and Džeroski (2004) also use an interleaving strategy and compares its influence when it is supplied at different frequencies.

In other works the advice takes the form of a reward. In this case, the teacher judges the quality of the agent's behavior sending a feedback signal that can be mapped onto a scalar value (e.g. by pressing a button or verbal feedback of "good" and "bad") (Thomaz and Breazeal, 2006; Knox and Stone, 2009). In contrast to RL, a learner agent seeks to directly maximize the short-term reinforcement given by the teacher. Other works combine the reward function of the MDP and that provided by the teacher (Knox and Stone, 2010; Knox et al., 2011).

4.2 Risk-directed Exploration

In this trend the exploration process is carried out by taking into account a defined risk metric. Gehring and Precup (2013) defines a risk metric based on the notion of *controllability.* If a particular state (or state-action pair) yields a considerable variability in the temporal-difference error signal, it is less controllable. The authors compute the controllability of a state-action pair as defined in

$$C(s_t, a_t) \leftarrow C(s_t, a_t) - \alpha'(|\delta_t| + C(s_t, a_t)) \tag{13}$$

The exploration algorithm uses controllability as an exploration bonus, picking actions greedily according to $Q(s_t, a_t)+wC(s_t, a_t)$. In this way, the agent is encouraged to seek controllable regions of the environment. This approach is successfully applied to the helicopter hovering control domain (García and Fernández, 2012) used in the RL Competition. Law (2005) uses a risk metric to guide the exploration process. In this case, the measurement of risk for a particular action in a given state is the weighted sum of the entropy (i.e., the stochasticity of the outcomes of a given action in a given state) and normalized expected return of that action. The risk measure of an action, $U(s, a)$, is combined with the action value to form the *risk-adjusted utility* of an action, i.e., $p(1 - U(s_t, a_t)) + (1 - p)Q(s_t, a_t)$ where $p \in [0, 1]$. The first term measures the safety value of an action, while the second term measures the long term utility of that action. The *risk-adjusted utility* is replaced in the Boltzmann function instead of the Q-values in order to safely guide the exploration process. However, the main drawback of these approaches is that the mechanism of risk avoidance is achieved by learning the risk values of actions during learning, i.e., when the functions $C(s_t, a_t)$ and $U(s_t, a_t)$ are correctly approximated. But it would be desirable to prevent risk situations from the early steps in the learning process.

Finally, it is important to note that the approaches considered here are similar in spirit to those in section 3.3. As an example, if the controllability was added as a constraint, then it becomes a constrained optimization criterion and, hence, this approach should be included in Section 3.3. However, we would obtain a different algorithm with different results. The approaches considered here only introduce a bias in the exploration process without satisfying hard constraints or constraints with a fixed probability of violation. Instead, the approaches in Section 3.3 must fulfill all the given constraints (although it is considered a fixed probability of constraint violation).

5. Discussion and Open Issues

In this Section, we complete the study of the techniques surveyed in this paper by classifying them across different dimensions. Additionally, we summarize the main advantages and drawbacks for each group of techniques in order to define future work directions.

5.1 Characterization of Safe RL Algorithms

As highlighted in the previous sections, current approaches to Safe RL have been designed to address a wide variety of problems where the risk considered and its detection have a large variety of forms. Table 2 analyzes most of the surveyed approaches across four dimensions.

5.1.1 Allowed Learner

We distinguish various RL approaches used in Safe RL. The *model free* methods such as Q-Learning (Sutton and Barto, 1998) which learn by backing up experienced rewards over time. The *model-based* methods which attempt to estimate the true model of the environment by interacting with it. The *policy search* methods which directly modify a policy over time to increase the expected long-term reward by using search or other optimization techniques. Finally, the *relational* RL methods which use a different state/action representation (relational or first-order language).

Table 2 shows that most of the approaches correspond to *model-free* RL algorithms. *Model-based* approaches are also used but few such methods handle continuous or large state and action spaces (Abbeel and Ng, 2005) and they generally have trouble scaling to tasks with many state and action variables due to the curse of dimensionality. *Model-based* approaches demand run exploration policies until they have an accurate model of the entire MDP (or at least the *reachable* parts of it). This makes many *model-based* approaches require exhaustive exploration that can take an undesirably long time for complex systems. Additionally, an aggressive exploration policy in order to build an accurate model can lead to catastrophic consequences. Therefore, *model-based* approaches suffer a similar exploration problem as *model-free* approaches, but in this case the question is: how can we safely explore the relevant parts of the state/action spaces to build up a sufficiently accurate dynamics model from which derive a good policy? Abbeel (2008) offers a solution to these problems by learning a dynamics model from teacher demonstrations. *Policy search* and *Relational* RL methods are also identified as techniques that can be applied to risky domains, but usually refer to the use of bootstrapping approaches.

5.1.2 Space Complexity

The column entitled by *Space* in Table 2 describes the complexity of the state and action spaces of the domains where the method has been used. The S refers to continuous or large

Citation	Allowed Learner	Spaces	Risk	Exploration
Modifying the Optimization Criterion: Section 3				
Worst Case Criterion: Section 3.1				
Heger (1994b)	model free	s/a	Var	$greedy$
Gaskett (2003)	model free	s/a	Var	$greedy$
Risk-Sensitive Criterion: Section 3.2				
Borkar (2002)	model free	s/a	Var	$greedy$
Mihatsch and Neuneier (2002)	model free	S/a	TD-error	$\epsilon - greedy$
Campos and Langlois (2003)	model free	S/a	TD-error	$\epsilon - greedy$
Geibel and Wysotzki (2005)	model free	S/a	error states	$greedy$
Constrained Criterion: Section 3.3				
Geibel (2006)	model free	s/a	$E(R) \geq \alpha$	greedy
Castro et al. (2012)	model free	S/A	$Var(R) \leq \alpha$	softmax
Moldovan and Abbeel (2011, 2012a)	model based	s/a	ergodicity	bonuses
Abe et al. (2010)	model free	s/a	ad-hoc constraints	greedy
Modifying the Exploration Process: Section 4				
Providing Initial Knowledge: Section 4.1.1				
Driessens and Džeroski (2004)	relational	S/a	initial exploration	softmax
Smart and Kaelbling (2000)	model free	S/A	initial exploration	gaussian
Martín H. and Lope (2009)	policy search	S/A	initial exploration	evolving NN
Koppejan and Whiteson (2011)	policy search	S/A	initial exploration	evolving NN
Maire (2005)	model free	s/a	initial exploration	$greedy$
Deriving a Policy from a Finite Set of Demonstrations: Section 4.1.2				
Abbeel and Ng (2005)	model based	S/A	accurate model	$greedy$
Tang et al. (2010)	model based	S/A	accurate model	$greedy$
Using Teacher Advice: Section 4.1.3				
Clouse (1997)	model free	S/a	similar $Q - values$	softmax
Hans et al. (2008)	model free	s/a	fatal transitions	level-based
García et al. (2013)	model free	S/A	unknown states	gaussian
Geramifard (2012)	model based	S/A	constrained states	softmax
Clouse and Utgoff (1992)	model free	s/a	human	$\epsilon - greedy$
Maclin and Shavlik (1996)	model free	s/a	human	softmax
Thomaz and Breazeal (2006, 2008)	model free	S/a	human	softmax
Walsh et al. (2011)	model based	s/a	$E(R) \leq \alpha$	R_{max}
Rosenstein and Barto (2002, 2004)	model free	S/A	agent/teacher confidence	gaussian
Kuhlmann et al. (2004)	model free	S/a	human	$\epsilon - greedy$
Torrey and Taylor (2012)	model free	S/a	agent/teacher confidence	$\epsilon - greedy$
Risk-directed Exploration: Section 4.2				
Gehring and Precup (2013)	model free	S/a	TD-error	risk directed
Law (2005)	model based	s/a	entropy and E(R)	risk directed

Table 2: This table lists most of the Safe RL methods discussed in this survey and classifies each in terms of four dimensions.

state space, and s to discrete and small state space. The same interpretation can be applied analogously to A and a in the case of the action space. In this way, S/a means that the method has been applied to domains with continuous or large state space and discrete and small action space.

Most of the research on RL has studied solutions to finite MDPs. On the other hand, learning in real-world environments requires handling with continuous state and action spaces. While several studies have focused on problems with continuous states, little attention has been paid to tasks involving continuous actions. These conclusions can also be obtained for Safe RL from Table 2 where most of the approaches address finite MDPs (Heger, 1994a; Gaskett, 2003; Moldovan and Abbeel, 2012a) and problems with continuous or large state spaces (Mihatsch and Neuneier, 2002; Geibel and Wysotzki, 2005; Thomaz and Breazeal, 2008), and much fewer approaches address problems with continuous or large state and action spaces (Abbeel et al., 2009; García and Fernández, 2012).

5.1.3 Risk

The forms of risk considered in this survey are also listed in Table 2. These forms are related to the variance of the return or its worst possible outcome (entitled *Var* in Table 2), to the temporal differences (entitled *TD-error*), to *error states*, to constraints related to the expected return (entitled by $E(R) \geq \alpha$ or $E(R) \leq \alpha$) or the variance of the return (entitled $Var(R) \leq \alpha$), to the *ergodicity* concept, to the effects of initial exploration in early stages in unknown environments (entitled *initial exploration*), to the obtaining of accurate models used later to derive a policy (entitled *accurate model*), to *similar Q-values*, to *fatal transitions*, to *unknown states*, to human decisions which determine what is considered a risk situation and when to provide help (entitled *human*), and to the degree of confidence both the teacher and the agent (denoted by *agent/teacher confidence*).

Table 2 shows the wide variety of forms of risk considered in the literature. This makes the drawing up of a benchmark problem difficult, or the identification of an environment to test different notions of risk. That is, in most cases, the approaches have different safety objectives and, hence, they result in different safe policies. For instance, the $\hat{Q} - Learning$ algorithm by Heger (1994b) leads to a safe policy completely different from that obtained by the method proposed by García and Fernández (2012). The applicability of one or another depends on the particular domain we are considering, and the type of risk it involves. Additionally, it is important to note that, in some cases, the risk metric selected places restrictions on which RL algorithm is used. For instance, the risk related to the *ergodicity* requires the model of the MDP to be known or learned.

5.1.4 Exploration

Table 2 also describes the exploration/exploitation strategy used for action selection. The *greedy* strategy is referred to the $\epsilon - greedy$ strategy where the ϵ is fixed at 0. For instance, Gaskett (2003) applies this exploration strategy and uses the inherent stochasticity of the environment to explore efficiently. In the classic $\epsilon - greedy$ action selection strategy the agent selects a random action with chance ϵ and the current best action with probability $1 - \epsilon$. In *softmax* action selection, the action probabilities are ranked according to their value estimates. The *gaussian* exploration is related to continuous action spaces and at every moment the action is selected by sampling from a Gaussian distribution with the mean at the current best action. The evolution of Neural Networks (NN) are used in policy search methods to explore around the policy space. In the *level-based* exploration proposed by Hans et al. (2008), the agent tries to explore all actions considered safe (i.e., it does not lead to a fatal transition) for each state gradually. The *exploration bonuses* adds a bonus to states with higher potential of learning (Baranes and Oudeyer, 2009), or with higher uncertainty (Brafman and Tennenholtz, 2003). Regarding the latter, the R_{max} exploration is related to model-based approaches and it divides states into two groups, known and unknown states, and focuses on reaching unknown states by assigning them maximum possible values (Brafman and Tennenholtz, 2003). Finally, the *risk directed* exploration uses a risk metric to guide the exploration process.

As the most widely used methods are *model free* in discrete and small action space, the exploration strategies most commonly used are $\epsilon - greedy$ and *softmax*. However, due to their random component of action selection, there is a certain chance of exploring dangerous or undesirable states. This chance affects the approaches differently using these strategies in Section 3 and Section 4. Most of the approaches in Section 3 are not interested in obtaining a safe exploration during the learning process; they are more interested in obtaining a safe policy at the end (Heger, 1994b; Gaskett, 2003; Mihatsch and Neuneier, 2002; Campos and Langlois, 2003). Therefore, the random component of these strategies is not so relevant for these approaches. In the approaches in Section 3.3 the use of these exploration strategies is limited to the space considered safe (i.e., that fulfills the constraints) (Geibel, 2006; Castro et al., 2012; Abe et al., 2010), which limits visiting undesirable regions despite this random component. Instead, most of the approaches in Section 4 address the problem of safe exploration using these exploration strategies. The approaches in Section 4.1.1 introduce an initial bias into the exploration space which mitigate (but do not prevent) the number of visits to undesirable states that produce these exploration strategies (Driessens and Džeroski, 2004; Maire, 2005). The approaches in Section 4.1.2 derive a model from a finite set of demonstrations, and then use this model to derive a policy greedily in an off-line and, hence, safe manner (Abbeel et al., 2009; Tang et al., 2010). The approaches in Section 4.1.3 combine the advice provided by the teacher with these exploration strategies to produce a safe exploration process (Clouse, 1997; Maclin et al., 2005a; Thomaz and Breazeal, 2006, 2008; Torrey and Taylor, 2012). For instance, the *softmax* exploration is used by Thomaz and Breazeal (2008) to select an action if no advice is introduced by the human. Otherwise, it selects a random action from among that derived from the advice introduced.

Other exploration strategies based on exploration bonuses such as R_{max} are related to the use of model-based algorithms (Walsh et al., 2011; Moldovan and Abbeel, 2012a). This exploration technique was first presented for finite MDPs (Brafman and Tennenholtz, 2003), but also there are versions for continuous state space where the number of samples required to learn an accurate model increases as the number of dimensions of the space grows (Nouri, 2011). Moldovan and Abbeel (2012a) use an adapted version of R_{max} where the exploration bonus of moving between two states is proportional to the number of neighboring unknown states that would be uncovered as a result of the move. It is important to note that R_{max} exploration by itself may be considered unsafe since it is encouraged to explore areas of *unknown* space, and other authors establish a direct relationship between the risk and the *unknown* concept (García and Fernández, 2012). However, Moldovan and Abbeel (2012a) use this exploration method to explore from among the policies whose preserve the *ergodicity* and are considered safe.

The gaussian exploration also introduces a random component in the algorithms proposed by García and Fernández (2012) and Smart and Kaelbling (2000). As regards the former, when an *unknown* state is found, the action is carried out by the teacher, otherwise, small amounts of Gaussian noise are randomly added to the greedy actions of the current policy. This ensures a safe exploration. As regards the latter, at the beginning of the learning process, the gaussian noise is added to greedy actions derived from a policy previously bootstrapped by teacher demonstrations. Other approaches uses a risk metric to direct the safe exploration based on the *temporal differences* (Gehring and Precup, 2013) or the weighted sum of an entropy measurement and the expected return (Law, 2005). Hans et al. (2008) uses a level-based exploration approach where the safe exploration is carried out gradually by exploring all the considered safe actions for each state. This exploration approach seems suitable for finite MDPs, but in MDPs with large state and action spaces, this exhaustive exploration is computationally intractable. Finally, the safe exploration conducted by Martín H. and Lope (2009) and Koppejan and Whiteson (2011) is due to the *population seeding* of the initial population which biases the subsequent exploratory process.

5.2 Discussion

Table 3 summarizes the main advantages and drawbacks of the approaches surveyed in this paper. Attending to the main advantages and drawbacks identified for each approach, we believe that there are several criteria that must be analyzed when developing Safe RL algorithms and risk metrics.

5.2.1 Selection of the Risk Metric

The algorithms based on the variance of the return (Sato et al., 2002; Borkar, 2002; Osogami, 2012) or its worst possible outcome (Heger, 1994b; Coraluppi, 1997) are not generalizable to problems in which a policy with a small variance can produce a large risk. To clarify this statement, we have reproduced the example set out by Geibel and Wysotzki (Geibel and Wysotzki, 2005). The example is a grid-world problem in which there are error states (i.e., undesirable or dangerous situations), and two goal states (one of them is placed next to the error state, and the other in a safer part of the state space). This grid-world is detailed in Figure 4.

The agent is able to move North, South, East, or West. With a probability of 0.21, the agent is not transported to the desired direction but to one of the three remaining directions. The agent receives a reward of 1 if it enters a goal state and a reward of 0 in every other case. There is no explicit negative reward for entering an error state, but when the agent enters it, the learning episode ends. In this domain, we found that a policy leading to the error states as fast as possible does not have a higher variance than one that reaches the goal states as fast as possible. Therefore, a policy with a small variance can therefore have a large risk, because this policy can lead the agent to error states. Additionally, we can see that all policies have the same worst case outcome and, hence, this optimization

Advantages	**Drawbacks**
Modifying the Optimization Criterion: Section 3	
Worst Case Criterion: Section 3.1	
• Useful when avoiding rare occurrences of large negative return is imperative	• Overly pessimistic • Variance of the return not generalizable to arbitrary domains • The true long term utility of the actions are lost • Not detect risky situations from the early steps
Risk-Sensitive Criterion: Section 3.2	
• Easy switch between risk-averse and risk-seeking behavior • Detect long-term risk situations	• If a conservative criterion is used, the policy may be overly pessimistic • The true long term utility of the actions are lost • Not detect risky situations from the early steps
Constrained Criterion: Section 3.3	
• Intuitively it seems a natural solution to the problem of safe exploration: the exploration is carried out only in the region of space considered safe (i.e., that fulfills the constraints)	• Many of these problems are computationally intractable, which difficult the formulation of RL algorithms • Correct selection of the parameter constraints • Constraints related to the return or its variance are not generalizable to arbitrary domains
Modifying the Exploration Process: Section 4	
Providing Initial Knowledge: Section 4.1.1	
• Bootstrap the value function approximation and lead the agent through the more relevant regions of the space from the earliest steps of the learning process	• Bias introduced may produce suboptimal policies • Exploration process following the initial training phase can result in visiting catastrophic states • Difficult initialization in complex structures
Deriving a Policy from a Finite set of Demonstrations: Section 4.1.2	
• The learning algorithm derives a policy from a finite set of demonstrations in an off-line and, hence, safe manner	• Learner performance is heavily limited by the quality of the teacher's demonstrations • How the agent should act when it encounters a state for which no demonstration exists?
Using Teacher Advice: Section 4.1.3	
• Guide the exploration process keeping the agent far away from catastrophic states from the earliest steps of the learning process • In ask for help approaches – Automatic detection of risk and request for advice when needed – Generalizable mechanisms of risk detection	• In ask for help approaches – Detect short term risk situations but not long term • In teacher provides advices approaches: – Teacher decides when to provide information based on its own feelings – Constant monitoring of the learner agent by the teacher might not be desirable in practice
Risk-directed Exploration: Section 4.2	
• Detect long-term risk situations	• Not detect risky situations from the early steps

Table 3: This table lists the main advantages and drawbacks of the Safe RL methods discussed in this survey.

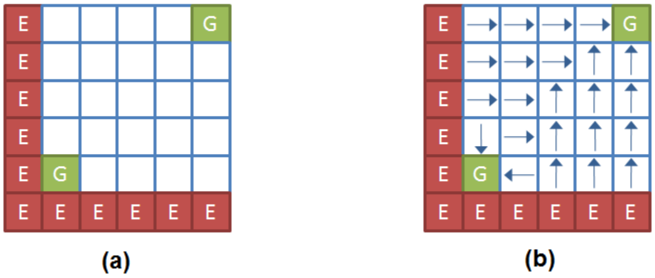


Figure 4: (a) The grid-world domain. (b) The minimum risk policy computed by Geibel and Wysotzki (Geibel and Wysotzki, 2005).

criterion is also unsuitable for this kind of risky domain. Accordingly, the variance or the worst-outcome criterion may not be generalizable to any risky domain.

The risk metric considered should be easily generalizable to any risky domain and be independent of the nature of the task. The risk based on the level of knowledge of a particular state is an example of generalizable risk metric (Hailu and Sommer, 1998; García and Fernández, 2012; Chernova and Veloso, 2009; Torrey and Taylor, 2012). This level of knowledge about a state can be based on the distance between the known and the unknown space (Hailu and Sommer, 1998; García and Fernández, 2012), on the difference between the highest and lowest Q-values (Clouse, 1997), or on the number of times an agent has made a non-trivial Q-value update in a state (Torrey and Taylor, 2012). In this sense, other knowledge level metrics in a state proposed in the literature can be used in Safe RL to identify potentially catastrophic situations (e.g., those based on the number of times an agent visits a state (Kearns and Singh, 2002), or on the *knownness criterion* (Nouri and Littman, 2008). The study of risk metrics easily generalizable to arbitrary domains is still an open issue in Safe RL.

5.2.2 Selection of the Optimization Criterion

We distinguish five possible situations based on two criteria: (i) the kind of optimization criterion (long term optimization of risk or risk-neutral), and (ii) the kind of risk detection (immediate and/or long term risk detection).

- *Long term optimization of risk.* In this case, we are interested in maximizing a long term measurement which considers some form of risk. This is common to most of the works reviewed in Section 3 where the risk-averse behavior is induced by transforming the classic optimization criterion of RL by introducing a risk metric. However, it seems difficult to find an optimization objective which correctly models our intuition of risk awareness. In Section 3.1, most of the approaches are updated based on a conservative criterion and the resulting policy tends to be overly pessimistic. Something similar happens with the approaches in Section 3.2 based on the variance of the return. In addition, the transformation of the optimization criterion produces a distortion in the action values and the true long term utility of the actions are lost.

Finally, most of these approaches repeatedly visit risk situations until the optimization criterion is correctly approximated and, subsequently, avoid dangerous situations. As an example of the latter, the optimization criterion used by Geibel and Wysotzki (2005) helps to reduce the number of visits to error states once the risk function is approximated (Geibel and Wysotzki, 2005; García and Fernández, 2012). It could be interesting to avoid future risk situations since it can provide a margin of reaction before reaching a point where it is unavoidable to reach an error state.

- *Detection of immediate risk.* We are interested in detecting and reacting to immediate risk situations from the early steps of the learning process while the classic optimization criterion remains. This second one is related to the approaches in Section 4. The worst thing that can happen to an agent in an environment may have a high return in the long term, but fatal consequences in the short term. The Safe RL algorithm should incorporate a mechanism to detect and react to immediate risk by manifesting different risk attitudes, while leaving the optimization criterion untouched. The ability to assess the amount of immediate risk in any action allows one to make step-by-step tradeoffs between attaining and abandoning the goal (i.e., the maximization of the return) in order to ensure the safety of the agent, the learning system and any external entity. The teacher advice approaches presented in Section 4.1.3.1 and Section 4.1.3.2 are good examples of this property. In these approaches, when a risk situation is detected by the agent or the teacher, the teacher provides safe information to the agent to prevent fatal situations. The main drawback of most of these approaches is that the risk is detected on the basis of the current state (Clouse, 1997; Geramifard, 2012; García and Fernández, 2012), and it may be too late to react.

- *Long term optimization of risk and immediate risk.* We are interested in maximizing a long term measurement which considers some form of risk and, at the same time, detects and reacts to immediate risk situations from the early steps of the learning process. The two previous approaches can be integrated into a same Safe RL algorithm. As an example, Geibel's approach (Geibel and Wysotzki, 2005) can be combined with an approach based on the level of knowledge of the current state from Section 4.1.3.1. The learner agent can ask for help in little known (Torrey and Taylor, 2012) or unknown states (Hailu and Sommer, 1998; García and Fernández, 2012) mitigating the effects of risk situations from early steps of the learning process. At the same time, the exploration directed by Geibel's optimization criterion, which include the risk function, $\rho^\pi(s)$, ensures the selection of *safe* actions preventing long-term risk situations once the risk function is correctly approximated. The development of these Safe RL algorithms is an area open for future research.

- *Detection of long term risk.* We are interested in detecting long-term risk situations when the risk function is correctly approximated, but not in detect and react to immediate risk situations from the early steps in the learning process. In addition, the classic optimization criterion remains. This is related to the approaches in Section 4.2. In this case, a risk metric is used to guide the exploration of the state and action spaces in a risk-directed exploration process based on the controllability (Gehring and Precup, 2013) or on the entropy (Law, 2005). In these approaches, the value function is learned separately, so optimal values and policies can be recovered if desired at a later time. As regards the latter, it would be interesting to decouple Geibel's risk function based on error states, $\rho^{\pi}(s)$, from the value function. In this way, it would be possible to analyze the effect of considering risk as part of the objective function with respect to considering risk only for risk-directed exploration while the objective function remains.

- *Detection of immediate and long term risk.* In this case, we are interested in detecting long-term risk situations when the risk function is correctly approximated, and in detect and react to immediate risk situations from the early steps in the learning process. The combination of immediate and long term risk detection mechanisms is still an open issue in Safe RL. As an example, the approaches in Section 4.1.3.1 could be combined with the approaches in Section 4.2. The learner agent can ask for help in little known (Torrey and Taylor, 2012) or unknown states (Hailu and Sommer, 1998; García and Fernández, 2012), and when the risk metric is greater than a certain threshold (e.g., $C(s_t, a_t) \geq w$ or $\rho^{\pi}(s) \geq w$). The first helps to mitigate the effects of immediate risk, the second immediate risk situations to be prevented in the long term through the delegation of the action taking to an external teacher instead of the ongoing exploratory process. At the same time, the risk-directed exploration mitigates the selection of actions which bring the risk situations closer.

5.2.3 Selection of the Mechanism for Risk Detection

The mechanism for risk detection should be automatic and not based on the intuition of a human teacher. In most of the approaches in Section 4.1.3.2, the teacher decides when to provide information to the learner agent based on its own feelings, but there is no metric as to the best time to do it. It is important to be aware of the fact that this way of providing the information is highly non-deterministic, i.e., the same human teacher can give the agent information in certain situations but remain impassive in other scenarios that are very similar. Moreover, the teacher observer can change his/her mind as to what is risky or not while the agent is still learning.

5.2.4 Selection of the Learning Schema

Although *policy search* methods have been demonstrated to be good techniques for avoiding risky situations, their safe exploration was related to the incorporation of *teachers* in the initial population (Martín H. and Lope, 2009; Koppejan and Whiteson, 2011). The problem of extracting knowledge (e.g., on the *known space*) from the networks or their weights makes it almost impossible to incorporate mechanisms for safe exploration during the learning process. As regards *model-free* vs. *model-based* approaches, there is still an open debate within the RL community as to whether *model-based* or *model-free* could be shown to be clearly superior to the other. This debate can also be taken in Safe RL. *Model-based* methods have relative higher space and computational complexities and lower sample complexity than *model-free* methods (Strehl et al., 2006). In general, this prevents the use of *model-based* methods in large space and stochastic problems in which the approximation of an accurate model from which derive a good policy is not possible. However, recent *model-based* approaches have demonstrated successful handling with continuous state domains (Nouri, 2011; Hester and Stone, 2013). Having such a model is a useful entity for Safe RL: it allows the agent to predict the consequences of actions before they are taken, allowing the agent to generate virtual experience. Therefore, we consider that building models is an open issue and a key ingredient of research and progress in Safe RL. So far, as shown in Table 2, most of the Safe RL approaches are *model-free*.

5.2.5 Selection of the Exploration Strategy

The exploratory process is responsible for visits to undesirable states or risky situations but also for progressively improve the policies learned. Techniques such as that proposed in Section 4.1.2 do not require exploration, but only the exploitation of the learning model derived from the teacher demonstrations. However, without additional exploration, the policies learned are heavily limited by the teacher demonstrations. The methods in Section 3.3 carry out safe exploration from among the policies in the constrained space. This may be the more advisable intuitively, but many of these problems are computationally intractable, which makes the formulation of RL algorithms difficult for large space and stochastic tasks.

As regards the other strategies used, all of them lead to a risky behavior. In exploration methods such as R_{max} the algorithm still tends to end up generating (and using) exploration policies in its initial stage. Additionally, R_{max} follows the *optimism in the face of uncertainty* principle, which consists of assuming a higher return on the most unknown states. This *optimism* for reaching unknown states can produce an unsafe exploration, since other authors establish a direct relationship between unknown and dangerous situations (García and Fernández, 2012). Exploration methods such as $\epsilon - greedy$, *softmax*, or gaussian incorporates a random component which give rise to a certain chance of exploring dangerous or undesirable states. The risk-directed exploration conducted by the use of risk metrics requires the function to be correctly approximated beforehand to avoid risk situations. Therefore, we consider that if the exploration is carried out in the entire state and action spaces (i.e., without constraints restricting the explorable/safe space), whatever the exploration used, this should be carried out in combination with an automatic risk detection mechanism (able to detect immediate and/or long term risk situations from the early steps in the learning process) and abandoning the goal in order to ensure the safety of the agent (e.g., asking for help from a teacher). Finally, if the goal is to obtain a safe policy at the end, without worrying about the number of dangerous or undesirable situations that occur during the learning process, then the exploratory strategy used to learn this policy is not so relevant from a safety point of view.

6. Conclusions

In this paper we have presented a comprehensive survey on Safe Reinforcement Learning techniques used to address control problems in which it is important to respect safety constraints. In this survey, we have contributed with a categorization of Safe RL techniques. We first segment the Safe RL techniques into two fundamental trends: the approaches based on the modification of the optimization criterion, and those based on the modification of the exploration process. We use this structure to survey the existing literature highlighting the major advantages and drawbacks of the techniques presented. We present techniques created specifically to address domains with a diverse nature of risk (e.g., those based on the variance of the return (Sato et al., 2002), on error states (Geibel and Wysotzki, 2005), or on the *controllability* concept (Gehring and Precup, 2013), and others that have not been created for this purpose, but have shown that their application to these domains can be effective in reducing the number of undesirable situations. Most of these techniques are described in Section 4.1 where external knowledge is used. As regards the latter, different forms of initialization have been shown to reduce the number of helicopter crashes successfully (Martín H. and Lope, 2009; Koppejan and Whiteson, 2011), or the number of times the agent moves into an obstacle in a Grid-World domain (Song et al., 2012; Maire, 2005); deriving a policy from a finite set of safe demonstrations provided by a teacher have also been shown to be a safe way of learning policies in risky domains (Abbeel et al., 2010); finally, the effectiveness of using teacher advice to provide actions in situations identified as dangerous has recently been demonstrated (García and Fernández, 2012; Geramifard, 2012).

The current proliferation of robots requires that the techniques used for learning tasks are safe. It has been shown that parameters learned in simulation often do not translate directly to reality, especially as heavy optimization on simulation has been observed to exploit the inevitable simplification of the simulator, thus creating a gap between simulation and application that reduces the usefulness of learning in simulation. In addition, autonomous robotic controllers must deal with a large number of factors such as the mechanical system and electrical characteristics of the robot, as well as the environmental complexity. Therefore, it is important to develop learning algorithms directly applicable to robots such as Safe RL algorithms since it could reduce the amount of damage incurred and, consequently, allow the lifespan of the robots to be extended.

Although Safe RL has proven to be a successful tool for learning policies which consider some form of risk, there are still many areas open for research, several of which we have identified in Section 5. As an example, the techniques based on the use of a risk function (Geibel and Wysotzki, 2005; Law, 2005; Gehring and Precup, 2013) have demonstrated their effectiveness in preventing risky situations once the risk function is correctly approximated. However, it would be desirable to prevent the risk situations from the early steps in the learning process. In this way, teacher advice techniques can be used to incorporate prior knowledge, thus mitigating the effects of immediate risk situations until the risk function is correctly approximated.

Acknowledgments

This paper has been partially supported by the Spanish Ministerio de Economía y Competitividad TIN2012-TIN2012-38079 and FEDER funds, and by the Innterconecta Programme 2011 project ITC-20111030 ADAPTA.

References

Pieter Abbeel. *Apprenticeship Learning and Reinforcement Learning with Application to Robotic Control.* PhD thesis, Stanford, CA, USA, 2008. AAI3332983.

Pieter Abbeel and Andrew Y. Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

Pieter Abbeel, Adam Coates, Timothy Hunter, and Andrew Y. Ng. Autonomous autorotation of an rc helicopter. In *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 385–394. Springer Berlin Heidelberg, 2009.

Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotic Research*, 29(13):1608–1639, 2010.

Naoki Abe, Prem Melville, Cezar Pendus, Chandan K. Reddy, David L. Jensen, Vince P. Thomas, James J. Bennett, Gary F. Anderson, Brent R. Cooley, Melissa Kowalczyk, Mark Domick, and Timothy Gardinier. Optimizing debt collections using constrained reinforcement learning. In *Proceedings of the 16th international conference on Knowledge discovery and data mining*, pages 75–84, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1.

Eitan Altman. Asymptotic properties of constrained markov decision processes. Rapport de recherche RR-1598, INRIA, 1992.

Brenna Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009. ISSN 09218890.

Drew Bagnell. *Learning Decisions: Robustness, Uncertainty, and Approximation.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2004.

Drew Bagnell and Jeff Schneider. Robustness and exploration in policy-search based reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 544–551, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

Drew Bagnell, Andrew Ng, and Jeff Schneider. Solving uncertain markov decision problems. Technical report, Robotics Institute Carnegie Mellon, 2001.

A. Baranes and P. Y. Oudeyer. R-IAC: Robust intrinsically motivated exploration and active learning. *Autonomous Mental Development, IEEE Transactions on*, 1(3):155–169, October 2009. ISSN 1943-0604.

Arnab Basu, Tirthankar Bhattacharyya, and Vivek S. Borkar. A learning algorithm for risk-sensitive cost. *Mathematics of Operational Research*, 33(4):880–898, 2008.

Vivek S. Borkar. A sensitivity formula for risk-sensitive cost and the actor-critic algorithm. *Systems & Control Letters*, 44:339–346, 2001.

Vivek S. Borkar. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27(2):294–311, May 2002. ISSN 0364-765X.

Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, March 2003. ISSN 1532-4435.

Andriy Burkov and Brahim Chaib-draa. Reducing the complexity of multiagent reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, page 44, 2007.

Pedro Campos and Thibault Langlois. Abalearn: Efficient self-play learning of the game abalone. In *INESC-ID, Neural Networks and Signal Processing Group*, 2003.

Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK*, 2012.

Victor Uc Cetina. Autonomous agent learning using an actor-critic algorithm and behavior models. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems, Estoril, Portugal*, pages 1353–1356, 2008.

Suman Chakravorty and David C. Hyland. Minimax reinforcement learning. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, USA, 2003.

Yin Chang-Ming, Han-Xing Wang, and Fei Zhao. Risk-sensitive reinforcement learning algorithms with generalized average criterion. *Applied Mathematics and Mechanics*, 28 (3):405–416, March 2007. ISSN 0253-4827.

Sonia Chernova and Manuela M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34:1–25, 2009.

Kun-Jen Chung and Matthew J. Sobel. Discounted mdps: distribution functions and exponential utility maximization. *SIAM Journal on Control Optimization*, 25(1):49–62, January 1987. ISSN 0363-0129.

Jeffery A. Clouse. On integrating apprentice learning and reinforcement learning. Technical report, Amherst, MA, USA, 1997.

Jeffery A. Clouse and Paul E. Utgoff. A teaching method for reinforcement learning. In *ML*, pages 92–110. Morgan Kaufmann, 1992. ISBN 1-55860-247-X.

Stefano P. Coraluppi. *Optimal control of markov decision processes for performance and robustness*. University of Maryland, College Park, Md., 1997.

Stefano P. Coraluppi and Steven I. Marcus. Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes. *Automatica*, 35:301–309, 1999.

Stefano P. Coraluppi and Steven I. Marcus. Mixed risk-neutral/minimax control of markov decision processes. *IEEE Transactions on Automatic Control*, 45(3):528–532, 2000.

Erick Delage and Shie Mannor. Percentile optimization for markov decision processes with parameter uncertainty. *Operations Research*, 58(1):203–213, January 2010.

Kurt Driessens and Sašo Džeroski. Integrating guidance into relational reinforcement learning. *Machine Learning*, 57(3):271–304, December 2004. ISSN 0885-6125.

Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan, May 2006.

Fernando Fernández, Javier García, and Manuela M. Veloso. Probabilistic policy reuse for inter-task transfer learning. *Robotics and Autonomous Systems*, 58(7):866–871, 2010.

Javier García and Fernando Fernández. Safe reinforcement learning in high-risk tasks through policy improvement. In *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning*, pages 76–83. IEEE, 2011.

Javier García and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, December 2012.

Javier García, Daniel Acera, and Fernando Fernández. Safe reinforcement learning through probabilistic policy reuse. In *Proceedings of the 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making*, October 2013.

Chris Gaskett. Reinforcement learning under circumstances beyond its control. In *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation*, 2003.

Clement Gehring and Doina Precup. Smart exploration in reinforcement learning using absolute temporal difference errors. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, Saint Paul, MN, USA*, pages 1037–1044, 2013.

Peter Geibel. Reinforcement learning for mdps with constraints. In *Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany,*, volume 4212 of *Lecture Notes in Computer Science*, pages 646–653. Springer, 2006. ISBN 3-540-45375-X.

Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

Alborz Geramifard. *Practical Reinforcement Learning using Representation Learning and Safe Exploration for Large Scale Markov Decision Processes.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, February 2012.

Alborz Geramifard, Joshua Redding, Nicholas Roy, and Jonathan P. How. UAV cooperative control with stochastic risk models. In *Proceedings of the American Control Conference*, pages 3393 – 3398, June 2011.

Alborz Geramifard, Joshua Redding, and JonathanP. How. Intelligent cooperative control architecture: A framework for performance improvement using safe learning. *Journal of Intelligent & Robotic Systems*, 72(1):83–103, 2013. ISSN 0921-0296.

Abhijit Gosavi. Reinforcement learning for model building and variance-penalized control. In *Proceedings of the Winter Simulation Conference*, pages 373–379. WSC, 2009.

Getachew Hailu and Gerald Sommer. Learning by biasing. In *Proceedings of the International Conference on Robotics and Automation*, pages 2168–2173. IEEE Computer Society, 1998. ISBN 0-7803-4301-8.

Alexander Hans, Daniel Schneegass, Anton M. Schäfer, and Steffen Udluft. Safe Exploration for Reinforcement Learning. In *Proceedings of the European Symposium on Artificial Neural Network*, pages 143–148, 2008.

Matthias Heger. *Risk and reinforcement learning: concepts and dynamic programming.* ZKW-Bericht. ZKW, 1994a.

Matthias Heger. Consideration of risk in reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 105–111, 1994b.

Alfredo García Hernández-Díaz, Carlos A. Coello Coello, Fatima Perez, Rafael Caballero, Julián Molina Luque, and Luis V. Santana-Quintero. Seeding the initial population of a multi-objective evolutionary algorithm using gradient-based information. In *Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, China*, pages 1617–1624, 2008.

Todd Hester and Peter Stone. TEXPLORE: Real-time sample-efficient reinforcement learning for robots. *Machine Learning*, 90(3), 2013.

Ronald A. Howard and James E. Matheson. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, 1972.

Marcus Hutter. Self-optimizing and pareto-optimal policies in general environments based on bayes-mixtures. In *Proceedings of the 15th Annual Conference on Computational Learning Theory*, Sydney, Australia, 2002.

Roberto Iglesias, Carlos V. Regueiro, J. Correa, E. Sanchez, and Senen Barro. Improving wall following behaviour in a mobile robot using reinforcement learning. In *Proceedings of the International symposium on engineering of intelligent systems*, Tenerife (España), February 1998a. ISBN 3-906454-12-6.

Roberto Iglesias, Carlos V. Regueiro, J.Correa, and Senen Barro. Supervised reinforcement learning: Application to a wall following behaviour in a mobile robot. In *Methodology and tools in knowledge-based systems*, pages 300–309, Castellon (España), June 1998b. Lecture notes in artificial intelligence 1415. ISBN 3-540-64574-8.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30:257–280, 2004.

Koen Hermans Jessica Vleugel, Michelle Hoogwout and Imre Gelens. Reinforcement learning with avoidance of unsafe regions. *BSc Project*, 2011.

Guofei Jiang, Cang-Pu Wu, and George Cybenko. Minimax-based reinforcement learning with state aggregation. In *Proceedings of the 37th IEEE Conference on Decision & Control*, Tampa, Florida, USA, 1998.

Kshitij Judah, Saikat Roy, Alan Fern, and Thomas G. Dietterich. Reinforcement learning via practice and critique advice. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence, Atlanta, Georgia, USA*, 2010.

Yoshinobu Kadota, Masami Kurano, and Masami Yasuda. Discounted markov decision processes with utility constraints. *Computers & Mathematics with Applications*, 51(2): 279–284, 2006.

Hisashi Kashima. Risk-sensitive learning via minimization of empirical conditional value-at-risk. *IEICE Transactions*, 90-D(12):2043–2052, 2007.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002. ISSN 0885-6125.

W. Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: the tamer framework. In *Proceedings of the 5th International Conference on Knowledge Capture*, September 2009.

W. Bradley Knox and Peter Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems*, May 2010.

W. Bradley Knox, Matthew E. Taylor, and Peter Stone. Understanding human teaching modalities in reinforcement learning environments: A preliminary report. In *Proceedings of the Agents Learning Interactively from Human Teachers Workshop*, July 2011.

Rogier Koppejan and Shimon Whiteson. Neuroevolutionary reinforcement learning for generalized helicopter control. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 145–152, July 2009.

Rogier Koppejan and Shimon Whiteson. Neuroevolutionary reinforcement learning for generalized control of simulated helicopters. *Evolutionary Intelligence*, 4:219–241, 2011.

Gregory Kuhlmann, Peter Stone, Raymond J. Mooney, and Jude W. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *Proceedings of the AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*, July 2004.

Edith L.M. Law. *Risk-directed exploration in reinforcement learning.* McGill University, 2005.

Long Ji Lin. Programming robots using reinforcement learning and teaching. In *Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14-19, 1991, Volume 2*, pages 781–786, 1991.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4):293–321, 1992.

Yaxin Liu, Richard Goodwin, and Sven Koenig. Risk-averse auction agents. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, pages 353–360. ACM, 2003. ISBN 1-58113-683-8.

David G. Luenberger. *Investment science.* Oxford University Press, Incorporated, 2013.

Richard Maclin and Jude W. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 22(1-3):251–281, 1996. doi: 10.1023/A:1018020625251.

Richard Maclin, Jude Shavlik, Lisa Torrey, Trevor Walker, and Edward Wild. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 2005a.

Richard Maclin, Jude Shavlik, Trevor Walker, and Lisa Torrey. Knowledge-based support-vector regression for reinforcement learning. In *Proceedings of the IJCAI'05 Workshop on Reasoning, Representation, and Learning in Computer Games*, 2005b.

Frederic Maire. Apprenticeship learning for initial value functions in reinforcement learning. In *Proceedings of the IJCAI'05 Workshop on Planning and Learning in A Priori Unknown or Dynamic Domains*, pages 23–28, 2005.

Shie Mannor and John N. Tsitsiklis. Mean-variance optimization in markov decision processes. In *Proceedings of the 28th International Conference on Machine Learning, Bellevue, Washington, USA*, pages 177–184, 2011.

Harry Markowitz. Portfolio selection. In *Journal of Finance*, volume 7, pages 77–91, 1952.

José Antonio Martín H. and Javier Lope. Learning autonomous helicopter flight with evolutionary reinforcement learning. In *Proceedings of the 12th International Conference on Computer Aided Systems Theory*, pages 75–82, 2009. ISBN 978-3-642-04771-8.

Helmut Mausser and Dan Rosen. Beyond var: From measuring risk to managing risk. *ALGO Research Quarterly*, 1(2):5–20, 1998.

John Mccarthy. Programs with common sense. In *Semantic Information Processing*, pages 403–418. MIT Press, 1959.

Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine Learning*, 49(2-3):267–290, 2002. ISSN 0885-6125.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *Proceedings of NIPS Workshop on Bayesian Optimization, Experimental Design and Bandits*, 2011.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK*, 2012a.

Teodor Mihai Moldovan and Pieter Abbeel. Risk aversion in markov decision processes via near optimal chernoff bounds. In *Advances in Neural Information Processing Systems 25, Lake Tahoe, Nevada, United States.*, pages 3140–3148, 2012b.

David L. Moreno, Carlos V. Regueiro, Roberto Iglesias, and Senen Barro. Using prior knowledge to improve reinforcement learning in mobile robotics. In *Proceedings fo the Conference Towards Autonomous Robotics Systems*, Bath (Reino Unido), September 2004.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning*, pages 799–806, 2010a.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Parametric return density estimation for reinforcement learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 368–375, Catalina Island, California, USA, Jul. 8–11 2010b.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operational Research*, 53(5):780–798, September 2005. ISSN 0030-364X.

Ali Nouri. *Efficient Model-Based Exploration in Continuous State-Space Environments.* PhD thesis, New Brunswick, NJ, USA, 2011. AAI3444957.

Ali Nouri and Michael L. Littman. Multi-resolution exploration in continuous spaces. In *Advances in Neural Information Processing Systems 21*, pages 1209–1216, 2008.

Takayuki Osogami. Robustness and risk-sensitivity in markov decision processes. In *Advances in Neural Information Processing Systems 25, Lake Tahoe, Nevada, United States*, pages 233–241, 2012.

Stephen D. Patek. On terminating markov decision processes with a risk-averse objective function. *Automatica*, 37(9):1379–1386, 2001.

Frederick Philip Klahr Hayes-Roth and David J. Mostow. Advice-taking and knowledge refinement: An iterative view of skill acquisition. *Cognitive Skills and Their Acquisition*, 1981.

Sameera S. Ponda, Luke B. Johnson, and Jonathan P. How. Risk allocation strategies for distributed chance-constrained task allocation. In *American Control Conference*, June 2013.

Martin L. Putterman. Markov decision processes: Discrete stochastic dynamic programming. Jhon Wiley & Sons, Inc, 1994.

Michael T. Rosenstein and Andrew G. Barto. Supervised learning combined with an actor-critic architecture. Technical report, Amherst, MA, USA, 2002.

Michael T. Rosenstein and Andrew G. Barto. *Supervised actor-critic reinforcement learning.* Wiley-IEEE Press, 2004.

Daniil Ryabko and Marcus Hutter. *Theorical Computer Science*, (3):274–284.

Makoto Sato, Hajime Kimura, and Shigenobu Kobayashi. TD algorithm for the variance of return and mean-variance reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence*, 16:353–362, 2002.

Nils T. Siebel and Gerald Sommer. Evolutionary reinforcement learning of artificial neural networks. *International Journal of Hybrid Intelligent Systems*, 4:171–183, August 2007. ISSN 1448-5869.

William D. Smart and Leslie Pack Kaelbling. Practical reinforcement learning in continuous spaces. In *Artificial Intelligence*, pages 903–910. Morgan Kaufmann, 2000.

Alice Smith, Alice E. Smith, David W. Coit, Thomas Baeck, David Fogel, and Zbigniew Michalewicz. *Penalty functions.* Oxford University Press and Institute of Physics Publishing, 1997.

Yong Song, Yi bin Li, Cai hong Li, and Gui fang Zhang. An efficient initialization approach of q-learning for mobile robots. *International Journal of Control, Automation and Systems*, 10(1):166–172, 2012.

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 881–888, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

Halit B. Suay and Sonia Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, pages 1–6. IEEE, July 2011. ISBN 978-1-4577-1571-6.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction.* The MIT Press, March 1998. ISBN 0262193981.

Giorgio Szegö. Measures of risk. *European Journal of Operational Research*, 163(1):5–19, 2005.

Hamdy A. Taha. *Operations research: an introduction.* Number 1. Macmillan Publishing Company, 1992. ISBN 9780024189752.

Aviv Tamar, Huan Xu, and Shie Mannor. Scaling Up Robust MDPs by Reinforcement Learning. *Computing Research Repository*, abs/1306.6189, 2013.

Jie Tang, Arjun Singh, Nimbus Goehausen, and Pieter Abbeel. Parameterized maneuver learning for autonomous helicopter flight. In *International Conference on Robotics and Automation*, 2010.

Matthew E. Taylor and Peter Stone. Representation transfer for reinforcement learning. In *Fall Symposium on Computational Approaches to Representation Change during Learning and Development*, November 2007.

Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.

Matthew E. Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(1):2125–2167, 2007.

Andrea L. Thomaz and Cynthia Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st National Conference on Artificial Intelligence*, AAAI'06, pages 1000–1005. AAAI Press, 2006. ISBN 978-1-57735-281-5.

Andrea Lockerd Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7): 716–737, 2008.

Lisa Torrey and Matthew E. Taylor. Help an agent out: Student/teacher learning in sequential decision tasks. In *Proceedings of the AAMAS Workshop Adaptive and Learning Agents*, June 2012.

Lisa Torrey, Trevor Walker, Jude Shavlik, and Richard Maclin. Using advice to transfer knowledge acquired in one reinforcement learning task to another. *Machine Learning: ECML 2005*, pages 412–424, 2005.

Paul E. Utgoff and Jeffrey A. Clouse. Two kinds of training information for evaluation function learning. In *Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14-19, 1991, Volume 2*, pages 596–600, 1991.

Pablo Quintía Vidal, Roberto Iglesias Rodríguez, Miguel Rodríguez González, and Carlos Vázquez Regueiro. Learning on real robots from experience and simple user feedback. *Journal of Physical Agents*, 7(1), 2013. ISSN 1888-0258.

Pradyot Korupolu VN and Balaraman Ravindran. Beyond rewards: Learning from richer supervision. In *Proceedings of the 9th European Workshop on Reinforcement Learning*, Athens Greece, September 2011.

Thomas J. Walsh, Daniel Hewlett, and Clayton T. Morrison. Blending autonomous exploration and apprenticeship learning. In *Proceedings of the Conference Advances in Neural Information Processing Systems 24, Granada, Spain*, pages 2258–2266, 2011.

Christopher Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge, UK, May 1989.

Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. ISBN 0-13-456567-3.