RISK-SENSITIVE DECISION MAKING VIA CONSTRAINED EXPECTED RETURNS

Jürgen Hahn *Abdelhak M. Zoubir*

Signal Processing Group
Technische Universität Darmstadt
Merckstraße 25, 64283 Darmstadt, Germany
E-mail: {jhahn, zoubir}@spg.tu-darmstadt.de

ABSTRACT

Decision making based on Markov decision processes (MDPs) is an emerging research area as MDPs provide a convenient formalism to learn an optimal behavior in terms of a given reward. In many applications there are critical states that might harm the agent or the environment and should therefore be avoided. In practice, those states are often simply penalized with a negative reward where the penalty is set in a trial-and-error approach. For this reason, we propose a modification of the well-known value iteration algorithm that guarantees that critical states are visited with a pre-set probability only. Since this leads to an infeasible problem, we investigate the effect of nonlinear and linear approximations and discuss the effects. Two examples demonstrate the effectiveness of the proposed approach.

Index Terms— Markov decision process, Risk, Decision making, Constrained optimization, Reinforcement Learning

1. INTRODUCTION

Decision making plays a crucial role in many applications today, e.g. in stock trading [1, 2], driving assistance [3, 4, 5], or communication [6, 7, 8, 9]. These problems can often be formulated as Markov decision processes (MDP) which provide a powerful statistical framework. Many algorithms have been developed to solve MDPs, e.g. value iteration [10] or Q-learning [11]. A key element of MDPs is the so-called reward function which provides the agent with rewards for a desired behavior. However, in practical applications it is often desired to avoid certain states which might harm the agent or its environment. This is especially important for robot control (e.g. avoidance of critical situations [12]) or driver assistance (e.g. collision avoidance [13]).

A simple solution is to penalize such states by providing (high) negative rewards. Though the agent will finally, after learning, attempt to avoid those states, this method raises two questions: firstly, what value should the penalty take? If the penalties are too high, they might distract the agent from the true target. Secondly, how certain is it that those states are really avoided? Considering the fact that the agent is acting in a stochastic environment, it is likely that the critical states cannot completely be avoided and, therefore, a quantifiable measure that can easily be interpreted is required which informs how likely it is that the agent is in a critical state.

In the reinforcement learning community the term risk is often related to the variance of the expected return [14, 15, 16]. Here, it is desired to obtain a value function that is invariant to small changes of the policy. In contrast, we define the term risk as the probability of visiting an undesired, critical state at any point in time similar to Geibel and Wysotzki [17]. They assume that the risk states are terminal states, meaning that the agent cannot recover and the process terminates. This allows them to formulate the problem of estimating the risk probabilities as another MDP. At the same time, they estimate the value function of the original MDP. Using a weighted sum of the risk and the original value function, the policy is estimated. A significant drawback of the algorithm in [17] is that the weight is *a priori* unknown and has to be searched for. Consequently, the MDPs have to be solved several times, leading to high computational costs. Further, this algorithm cannot be used in applications where the visit of a critical state does not lead to a termination of the process.

We propose the use of a standard value iteration algorithm whose maximization step is constrained to guarantee that the probability of visiting a critical state is below a pre-set threshold at any point in time. As this leads to an infeasible optimization problem with an infinite number of constraints, we present an approximation that can be solved by means of a nonlinear program. We further explain in which scenarios a linear approximation, similar to constrained MDPs (CMDP) [18, 19, 20], is useful. In contrast to [17], we only need to solve the MDP once, rendering this approach attractive for large state and action spaces. Further, we do not require critical states to be terminal states. In summary, our approach to control the risk is suitable if the problem at hand has the following properties:

- critical states are potentially unavoidable due to the stochastic nature of the system (e.g. failure of an engine) and are not necessarily terminal states
- there is usually more than one possibility to reach a goal

(otherwise the problem might be unsolvable)
- risk is not directly quantifiable as a cost (e.g. injury of people or failure of a machine)

In Section 2, we briefly revisit the Markov decision process as the underlying framework of the proposed algorithm. In Section 3, the problem is formulated and solutions are presented. The simulations in Section 4 demonstrate the performance of the proposed approach, before concluding.

2. MARKOV DECISION PROCESSES

A finite MDP with an infinite horizon is defined by

- a finite set of N states, $\mathcal{S} = \{s^{(1)}, s^{(2)}, \dots, s^{(N)}\}$
- a finite set of M actions, $\mathcal{A} = \{a^{(1)}, a^{(2)}, \dots, a^{(M)}\}$
- the initial state distribution $P(s_0)$
- the transition probability $P(s'|s,a)$ with $s' \in \mathcal{S}$
- the discount factor $\gamma \in [0,1)$
- the reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ with absolute value bounded by $R_{\max} \in \mathbb{R}$

The transition probability $P(s'|s,a)$ describes the probability of ending up in state s' when taking action a in state s. Thus, it provides information, for example, about the probability that the agent successfully performs an action or the agent ends up in a certain state due to system dynamics and plays therefore a crucial role for risk estimation.

In decision making, the goal is to find an action that maximizes the expected discounted return or value $V(s)$ for each state,

$$V(s|\pi) = \mathsf{E}[R(s_{t=0}) + \gamma R(s_{t=1}) + \gamma^2 R(s_{t=2}) + \dots |\pi],$$

i.e. to find an optimal policy π, $\pi : \mathcal{S} \to \mathcal{A}$. The expected return informs the agent about the collected reward to expect when acting according to the policy at time steps $t = 0, 1, 2, \dots$ and can be estimated by means of the Bellman equations [10, 21].

Though it has been shown that a deterministic policy as described above is optimal in many decision making problems [22], we assume the policy to be stochastic in the sequel, i.e. the decision maker chooses an action with a certain probability. This will be justified in Section 3.1. Since we consider finite state and action spaces, we model the policy π as a categorical distribution, i.e.

$$\pi := P_\theta(a|s) = \prod_{i=1}^{|\mathcal{A}|} \prod_{j=1}^{|\mathcal{S}|} \theta_{ij}^{1(a_i,a)1(s_j,s)}$$

with parameters $\theta_{ij}, 0 \le \theta_{ij} \le 1, i = 1, \dots, M, j = 1, \dots, N$.

3. RISK-SENSITIVE DECISION MAKING

In the context of decision making, we define the term 'risk' as the probability of visiting a critical state $s^{(c)} \in \mathcal{S}_c$ for each time step where $\mathcal{S}_c$ denotes the set of critical states. Our aim is to make sure that the probability of the agent being in a critical state is below some pre-set threshold α for all time steps t, $t = 1, 2, \dots$, assuming this is already fulfilled for the initial state distribution $P(s_0)$. Thus, we ideally consider the entire life span of the system.

The probability of visiting a certain state during time step t can be recursively computed given the policy following the Markov chain,

$$P_\theta(s_{t+1}) = \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} P(s_{t+1}|s_t, a_t) P_\theta(a_t|s_t) P(s_t). \quad (1)$$

Assuming that the critical states are always terminal states, the recursive structure can be exploited to derive a Q-learning based-scheme to calculate the probabilities of the critical states [17]. In contrast, we propose to constrain the maximization of the value-function $V(s)$ directly, since we want to make sure that, for any time step t, the risk of entering a critical state is bounded by the pre-set threshold α. In the infinite horizon case, i.e. when the process does not terminate, the maximization problem has then an infinite number of constraints,

$$\max_\pi V(s|\pi) \quad \text{s.t.}$$
$$P_\theta(s_t = s^{(c)}) \le \alpha \quad \forall s^{(c)} \in \mathcal{S}_c,\ t = 1, 2, \dots$$

and is therefore infeasible to solve. An algorithm for finding an approximate solution to this problem is presented in the sequel.

3.1. Risk-bounded value iteration

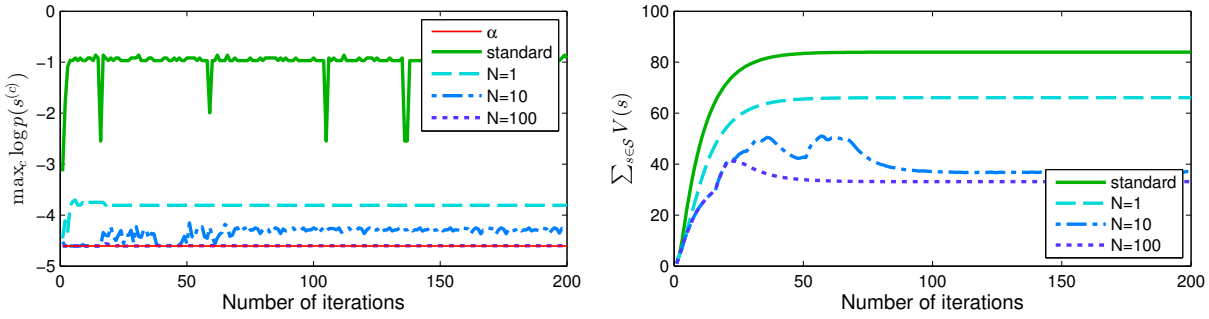
The proposed algorithm is basically a modification of the well-known value iteration algorithm. In principle, any value-based RL algorithm (such as Q-learning) can be modified to consider risks as defined in Section 3. Since we assume that the transition probability is known, value iteration is most appealing due to its simplicity and effectiveness.

The key idea is to exploit the Bellman-Equations, which allow to iteratively estimate the expected return. Each iteration contains two steps. In the first step, as in value iteration, the state-action value-function $Q(s,a)$ is estimated for all state and action pairs by adding the expected discounted return of the neighboring states,

$$Q(s,a) \leftarrow R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s')$$
$$\forall\, s \in \mathcal{S}, a \in \mathcal{A},$$

i.e. we assume $V(s)$ to be known and estimate the expected return for a single time step.

In the second, step we seek for an optimal probability distribution over the actions for each state. Here, the consideration of the risk comes into play: we aim at maximizing the state-action value function such that the risk is bounded by



(a) (b)

Fig. 1: Results of the robot navigation example. (a) Using standard value iteration, the probability of the risk is rather high. Considering only the first-order constraints, the risk is significantly reduced but still exceeds the requested threshold. (b) Avoiding critical states leads to a reduction of the expected return. The fluctuations in the range from 20 to 70 iterations are caused by numerical issues of the solver.

the threshold α and, thus, obtain an update of $V(s)$. This is formulated as a constrained optimization problem:

$$\begin{aligned} &\max_{\theta} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} Q(s,a) P_\theta(a|s) \\ \text{s.t.} \quad &P_\theta(s_t = s^{(c)}) \leq \alpha \quad \forall s^{(c)} \in \mathcal{S}_c,\ t = 1, 2, \dots \end{aligned} \tag{2}$$

If the underlying Markov chain is irreducible, $P_\theta(s_t = s^{(c)})$ will converge to the steady state distribution for $t \to \infty$. In this case, and if N is chosen sufficiently high, computing the constraints up to order N, i.e. $t \in [1, N]$, yields a good approximation. Nonetheless, this problem is nonlinear due to the constraints for $t > 1$ and is therefore difficult to solve.

In standard value iteration, it has been shown that a deterministic policy is optimal [22]. This does not hold for the constrained problem as can be seen in Eq. (2). Here, we allow the agent to choose actions leading to critical states as long as the constraints are fulfilled and those states result in higher rewards. However, we are able to limit the risk by reducing the probability of taking these actions. In contrast, if a deterministic policy was chosen, the agent would always have to perform the safer actions, leading to a potentially much lower expected return.

3.2. Linear approximation: first-order constraint

In many applications, it is desired to avoid critical states. Thus, the threshold α will be set rather low such that the agent shall not be directed to risk state, i.e. the policy is pointing to some other state. This can be achieved by fulfilling the first order constraints, i.e. $P(s_1 = s^{(c)}) \leq \alpha\ \forall s^{(c)} \in \mathcal{S}_c$. Consequently, ending up in a critical state can only be caused by system dynamics or unexpected behavior of the system.

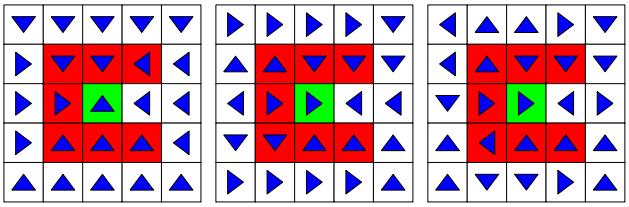
In this scenario, we can avoid the expensive nonlinear optimization since the problem in Eq. (2) becomes linear and can easily be solved by a linear program. Thus, we even obtain a globally optimal solution.

4. SIMULATIONS

To show the effects of different orders N, we consider two examples. The first example is based on the well-known grid-world [21]. The second one investigates the effect in a highly dynamical environment.

4.1. Robot navigation

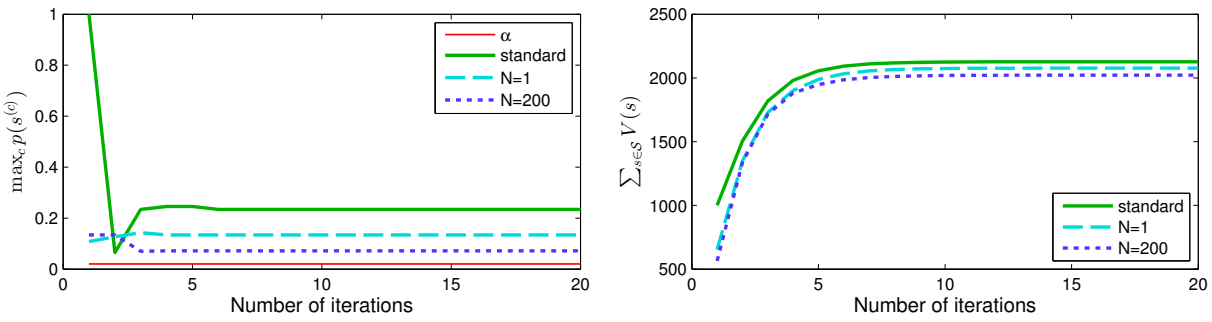
Consider a robot moving in a maze. The states are given by the location of the robot and its possible actions are moving east, west, north or south. We assume a non-ideal robot, meaning that with a probability of 20%, it randomly ends up in a neighboring state instead of the intended one. Whenever an action leads to hitting a wall, it will remain in the current state. Since a navigation problem is simulated, we place a positive reward (+1) in the center of the maze as indicated by the green field in Fig. 2, otherwise the reward is zero. The initial state distribution is uniform.



(a) Standard (b) $N = 1$ (c) $N = 100$

Fig. 2: Policies for the robot example. Only the action maximizing $P(a|s)$ is shown for each state.

We require that the probability of visiting critical states indicated by the red color in Fig. 2 should not exceed the threshold $\alpha = 1\%$. Fig. 1a shows that using order $N = 100$ ensures that the risk is not exceeded. The first-order approximation exceeds the threshold slightly, but the estimated risk is



(a) (b)

Fig. 3: Results of the machine example. (a) Only by means of high order constraints it is possible to guarantee low risks close to the desired threshold. (b) Though the risk is significantly reduced using the risk-sensitive approaches, the expected return only slightly decreases.

still much lower than that of standard value iteration. Avoiding risk states reduces the expected return significantly as depicted in Fig. 1b. As shown in Fig. 2a, the standard value iteration algorithm does not avoid the critical states and finds the shortest path to the reward and has therefore the highest expected return and risk. The first-order risk-sensitive algorithm circumvents the critical states but finds the reward nonetheless (Fig. 2b). With order $N = 100$, the risk threshold is met and the policy directs the agent opposite of many critical states (Fig. 2c).

4.2. Machine replacement

In this example, we are interested in estimating the optimal point in time for replacing a machine that produces goods worth 100 per time step. However, the probability of failure during the production of an item grows exponentially with time, while the initial probability of failure is 10%. The cost of replacing the machine is 30. We formulate this problem as an MDP, where the state space is defined by the age of the machine and the number of items produced (successful or failed) and the initial state distribution is uniform. Two actions can be taken, either replacing (R) the machine or simply waiting (-). When a machine is replaced, the age is reset to one and the probability of success is 90% again. Otherwise, the age of the machine will increase and an item might be produced. We set $\gamma = 0.9$, i.e. we discount items produced at higher ages.

A failure of the machine is assumed to cause the entire process chain to stop, leading to monetary loss and a loss of trust by potential customers, making the cost of this failure difficult to quantify. Thus, we define all states where no item is produced as risk states (indicated in red color in Fig. 4) and set the threshold to $\alpha = 2\%$. Fig. 4a shows that according to the standard value iteration, the machine should be replaced after six time steps. As depicted in Fig. 3a, the risk exceeds the required threshold. The first-order approximation also cannot fulfill the constraints. Only considering an order $N = 200$ shows a risk that is in the range of the requested bounds. Though the risk has been significantly reduced, the expected returns only slightly decrease (Fig. 3a). The policies of the risk-sensitive approaches show that the machine should be earlier replaced - in case of the nonlinear one basically directly after the first time step, if no item has been produced.

Note that this example is particularly challenging in terms of risk reduction, since the described system underlies strong dynamics, meaning that the action of the agent has a comparably small effect only. Thus, the required threshold on the risk cannot be perfectly met.

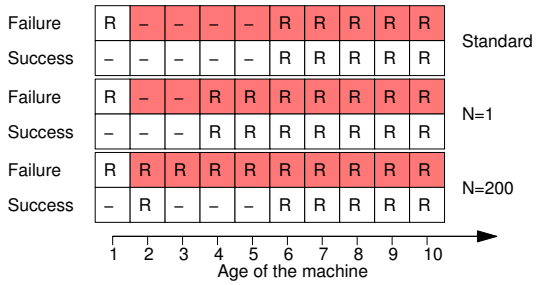


Fig. 4: Policies for the machine example. Only the action maximizing $P(a|s)$ is shown for each state.

5. CONCLUSION

We define the term risk as the probability of visiting a critical state for every time step. Our aim was to derive an algorithm that guarantees that the agent visits critical states with a probability less than a pre-set threshold, leading to an infeasible problem with an infinite number of constraints. Thus, we have provided insight to nonlinear and linear approximations and presented an algorithm based on the well-known value iteration algorithm. By means of two examples, we have shown that these approximations lead to useful results. Especially the linear approximation is fast to compute and provides good results as long as the system dynamics lead the agent into critical states with a low probability only.

6. REFERENCES

[1] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 875–889, Jul 2001.

[2] J. W. Lee, J. Park, J. O, J. Lee, and E. Hong, "A multiagent approach to Q-learning for daily stock trading," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, no. 6, pp. 864–877, Nov 2007.

[3] M. Liebner, F. Klanner, M. Baumann, C. Ruhhammer, and C. Stiller, "Velocity-based driver intent inference at urban intersections in the presence of preceding vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 10–21, Summer 2013.

[4] F. Muehlfeld, I. Doric, R. Ertlmeier, and T. Brandmeier, "Statistical behavior modeling for driver-adaptive precrash systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1764–1772, Dec 2013.

[5] J. Wang, L. Zhang, D. Zhang, and K. Li, "An adaptive longitudinal driving assistance system based on driver characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 1–12, Mar 2013.

[6] N. Ghasemi and S. Dey, "A constrained mdp approach to dynamic quantizer design for hmm state estimation," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1203–1209, March 2009.

[7] N. Mastronarde and M. van der Schaar, "Fast reinforcement learning for energy-efficient wireless communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6262–6266, Dec 2011.

[8] J. Lunden, S. R. Kulkarni, V. Koivunen, and H. V. Poor, "Multiagent reinforcement learning based spectrum sensing policies for cognitive radio networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 858–868, Oct 2013.

[9] A. Minasian, R.S. Adve, and S. Shahbazpanahi, "Energy harvesting for relay-assisted communications," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4763–4767.

[10] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

[11] C. J. C. H. Watkins, *Learning from Delayed Rewards*, Ph.D. thesis, Cambridge University, Cambridge, England, 1989.

[12] T. Sanger, "Risk-aware control," *Neural Computation*, vol. 26, no. 12, pp. 2669–2691, Dec 2014.

[13] J. Park, J. H. Yoon, M.-G. Park, and K.-J. Yoon, "Dynamic point clustering with line constraints for moving object detection in DAS," *IEEE Signal Processing Letters*, vol. 21, no. 10, pp. 1255–1259, Oct 2014.

[14] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine Learning*, vol. 49, no. 2-3, pp. 267–290, 2002.

[15] V. S. Borkar, "Q-learning for risk-sensitive control," *Mathematics of Operations Research*, vol. 27, no. 2, pp. 294–311, 2002.

[16] Y. Shen, M. J. Tobia, T. Sommer, and K. Obermayer, "Risk-sensitive reinforcement learning," *Neural Computation*, vol. 26, no. 7, pp. 1298–1328, 2014.

[17] P. Geibel and F. Wysotzki, "Risk-sensitive reinforcement learning applied to control under constraints.," *Journal of Artificial Intelligence Research (JAIR)*, vol. 24, pp. 81–108, 2005.

[18] E. Altman, *Constrained Markov decision processes*, vol. 7, CRC Press, 1999.

[19] D.V. Djonin and V. Krishnamurthy, "Q -learning algorithms for constrained markov decision processes with randomized monotone policies: Application to mimo transmission control," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2170–2181, May 2007.

[20] S. Feyzabadi and S. Carpin, "Risk-aware path planning using hirerachical constrained Markov decision processes," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2014, pp. 297–303.

[21] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, MIT Press, Cambridge, MA, USA, 1998.

[22] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.