Robust Kalman Temporal Difference

Shirli Di-Castro Shashua [1] SHIRLIDI@TX.TECHNION.AC.IL

Shie Mannor [1] SHIE@EE.TECHNION.AC.IL

Abstract

We propose an *on-line* algorithm for policy evaluation in large scale Robust Markov Decision Processes (RMDPs) with uncertainty in the transition probabilities. Our approach is based on the Kalman-Temporal Difference (KTD) formulation, supporting linear and non-linear approximations and considering minimal conditions on the uncertain transition probabilities. Previous work deals with robustness using dynamic programming (DP) and approximate dynamic programming (ADP) methods for both small and large state spaces. These methods can be used only in an off-line setting that requires full trajectories information. In large scale state spaces, the convergence proof is based on a restricted assumption regarding the uncertainty set and only linear value function approximation is considered. Our approach overcomes these limitations by using the Kalman filter framework for on-line estimation and considering the robust Bellman equation as an observation function. We present the Robust-KTD algorithm, analyze its convergence and examine its performance.

1. Introduction

Sequential decision processes in stochastic dynamic environments with uncertainty in the transition probabilities are often modeled as Robust Markov Decision Processes (RMDPs; Nilim and El Ghaoui, 2005; Iyengar, 2005). Given a state transition uncertainty set, the aim of an agent is to find the optimal policy which maximizes the worst case value function over the associated uncertainty set. This framework is useful when the transition probabilities are estimated, usually from noisy data, or when the agent asks for policies that are risk-sensitive or follows some coherent risk measurements (Petrik and Zilberstein, 2011; Chow et al., 2015). If the estimation error is not considered, it may cause a degradation in the performance of the chosen policy (Mannor et al., 2007). For small or medium state spaces, the solution for an RMDP can be obtained by dynamic programming (DP). Recently, an algorithm for solving RMDPs in dynamic environments with large state spaces has been developed by using approximate dynamic programming (ADP) approach with linear approximation (Tamar et al., 2014). Their convergence analysis is based on a restrictive assumption over the transitions in the uncertainty set and their approach requires a linear approximation of the value function and is not adapted to non-linear estimations. In both approaches, DP and ADP, the policy estimation procedure is done off-line, requiring a pass over full trajectories information.

This paper considers *on-line* policy estimation in large scale RMDPs which is based on the Kalman Temporal Difference (KTD) framework (Geist and Pietquin, 2010). In this framework the Kalman Filter method (Kalman, 1960) is used for on-line tracking and

1. Faculty of Electrical Engineering, Technion - Israel Institute of Technology, Haifa, Israel.

estimating the state in dynamic environments through indirect observations of this state. KTD uses temporal difference (TD) error observations for estimating the value function. We will consider the robust Bellman equation as a function that links the indirect observations to the value function estimation in RMDPs. This approach takes into consideration the uncertainty in the transition probabilities and allows minimal conditions on the uncertainty set, reducing the restrictive assumption that appears in Tamar et al. (2014). In addition, both linear and non-linear value function estimations can be used to evaluate the robust policy. Our contributions are a method for on-line estimation of the value function in RMDPs using linear or non-linear approximations, with convergence proof; performance examination in a small state space domain.

2. Background

In this section we describe our problem setting and preliminaries from RMDPs and KTD.

2.1 Robust Markov Decision Processes

For a discrete set $\mathcal{X}$, $\mathcal{M}(\mathcal{X})$ denotes the set of probability measures on $\mathcal{X}$. An RMDP (Iyengar, 2005; Nilim and El Ghaoui, 2005) is a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma\}$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a deterministic and bounded reward function, γ is a discount factor and $\mathcal{P}(s, a) \subset \mathcal{M}(\mathcal{S})$ is a probability uncertainty set for each state and action. We consider the infinite horizon case, where at each discrete time step i the system stochastically steps from state $s_i \in \mathcal{S}$ to state $s_{i+1} \in \mathcal{S}$ by taking an action $a_i \in \mathcal{A}$. Each transition (s_i, a_i, s_{i+1}) is associated with an immediate reward $r(s_i, a_i)$. The agent chooses the actions according to a policy $\pi : \mathcal{S} \to \mathcal{M}(\mathcal{A})$ that maps each state to a probability distribution over the actions set. The system then moves to a successive state, s_{i+1}, according to the probability distribution $P(s_i, a_i)$ which is assumed to lie in a *known* uncertainty set $\mathcal{P}$. This set can be obtained from confidence intervals on historical data. We adopt the Rectangularity property for $\mathcal{P}$ (Iyengar, 2005). In section 3 we elaborate on the this property and its relevance to our algorithm.

The value function (Q-function) of state s (state-action pair (s, a)) under policy π and state transition model P represents the expected sum of discounted returns when starting form state s (state-action pair (s, a)) and executing policy π (Sutton and Barto, 1998): $V^{\pi,P}(s) = \mathbb{E}^{\pi,P}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, a_i) | s_0 = s\right]$, $\left(Q^{\pi,P}(s, a) = \mathbb{E}^{\pi,P}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, a_i) | s_0 = s, a_0 = a\right]\right)$, where $\mathbb{E}^{\pi,P}$ denotes the expectation w.r.t the state-action distribution induced by the transitions P and the policy π. In RMDPs, we are interested in finding the policy that maximizes the *worst case* value function $V^{\pi}(s) = \inf_{P \in \mathcal{P}} V^{\pi,P}(s)$ or the *worst case* Q-function $Q^{\pi}(s, a) = \inf_{P \in \mathcal{P}} Q^{\pi,P}(s, a)$. The optimal robust value function is then: $V^*(s) = \sup_{\pi}\{\inf_{P \in \mathcal{P}} V^{\pi,P}(s)\}$.

As proved in Iyengar (2005) and Nilim and El Ghaoui (2005), the robust optimal value function, $V^*(s)$, is the unique solution of the Bellman recursion: $V^*(s) = \sup_{a \in \mathcal{A}}\{r(s, a) + \gamma \inf_{P \in \mathcal{P}} \mathbb{E}^P\left[V^*(s') | s, a\right]\}$, where s' is the successive state when taking an action a in state s. Iyengar (2005) showed that the agent can be restricted to stationary deterministic Markov

policies without affecting the achievable robust reward. We therefore restrict ourselves to this family of policies.

The procedure for finding the optimal policy can be based on policy iteration (PI) scheme where the algorithm repeatedly improves policy π according to policy evaluation steps:

$$\begin{cases} V^\pi(s) = r(s,\pi(s)) + \gamma \inf_{P\in\mathcal{P}} \mathbb{E}^P\left[V^\pi(s')|s,\pi(s)\right] \\ Q^\pi(s,a) = r(s,a) + \gamma \inf_{P\in\mathcal{P}} \mathbb{E}^P\left[Q^\pi(s',\pi(s'))|s,a\right] \end{cases} . \tag{1}$$

Then the policy improvement step updates the current policy to a better policy by choosing actions that optimize the Q-function for each state: $\pi(s) = \arg\max_a Q^\pi(s,a)$. Our algorithm presented in Section 3 is used in policy evaluation steps. The policy improvement step can be done by any preferable scheme (Howard, 1960; Konda and Borkar, 1999).

2.2 Kalman Filter for Temporal Difference

The KTD framework (Geist and Pietquin, 2010) presents an alternative point of view of the reinforcement learning problem and its formulation under the MDP framework. The parameter vector $\boldsymbol{\theta}$ that represents the value function $\hat{V}_{\boldsymbol{\theta}}(s)$ or the Q-function $\hat{Q}_{\boldsymbol{\theta}}(s,a)$ is not concerned as a deterministic variable, but rather as a random variable that evolves as a random walk: $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} + \mathbf{v}_i$, where $\boldsymbol{\theta}_i$ is the value function estimation parameter at time i and $\mathbf{v}_i$ is a white stochastic evolution noise. The immediate reward r_i is treated as a local observation of the estimated parameter vector. In the TD approach (Sutton and Barto, 1998), the reward is associated to the parameter vector by the Bellman equation: $r_i = h(\boldsymbol{\theta}_i, t_i) + n_i$ where

$$t_i = \begin{cases} (s_i, s_{i+1}) \\ (s_i, a_i, s_{i+1}, a_{i+1}) \end{cases} , \quad h(\boldsymbol{\theta}_i, t_i) = \begin{cases} \hat{V}_{\boldsymbol{\theta}_i}(s_i) - \gamma \hat{V}_{\boldsymbol{\theta}_i}(s_{i+1}) \\ \hat{Q}_{\boldsymbol{\theta}_i}(s_i, a_i) - \gamma \hat{Q}_{\boldsymbol{\theta}_i}(s_{i+1}, a_{i+1}) \end{cases} \tag{2}$$

and n_i is a white stochastic measurement noise. From a sequence of such observations the agent updates its estimation of the parameter vector. This approach is driven from the Kalman filtering method that provides algorithms for tracking and estimating a state of possibly non-stationary dynamic system (Kalman, 1960).

The KTD framework presented so far assumes deterministic transitions. Geist and Pietquin (2010) provide an additional algorithm, called XKTD (eXtended KTD), for cases where the observation function $h(\boldsymbol{\theta}_i, t_i)$ may not be linear and the transition probabilities are stochastic. Geist and Pietquin (2010) were motivated by the colored noise model for the observation noise n_i, described in Engel et al. (2005). This noise model was driven from the Bellman evaluation equation and is presented in Assumption 1:

Assumption 1 *In the observation equation $r_i = h(\boldsymbol{\theta}_i, t_i) + n_i$, the observation noise n_i is assumed to be colored with a (scalar) variance P_{n_i}. The model describing the colored observation noise is the* First-Order Moving Average (MA) noise model*: $n_i = -\gamma u_i + u_{i-1}$, where u_i is a white random process, $u_i \sim (0, \sigma^2_{n_i})$.*

For accounting the colored noise model in the estimation process, Geist and Pietquin (2010) propose to express the scalar MA noise n_i as a vectorial Autoregressive (AR) noise denoted by $\mathbf{u}'_i = [-\gamma u_i, u_i]^\top$ with covariance matrix $\mathbf{P}_{\mathbf{u}'_i} = \mathbb{E}[\mathbf{u}'_i \mathbf{u}'^\top_i]$. This noise vector

can be augmented to the parameter vector for estimate them *jointly*, resulting the Kalman equations for the XKTD algorithm: $\begin{cases}\mathbf{x}_i = \mathbf{F}\mathbf{x}_{i-1} + \mathbf{v}'_i \\ r_i = h(\mathbf{x}_i, t_i)\end{cases}$, where $\mathbf{x}_i = (\boldsymbol{\theta}_i^\top, n_i, w_i)^\top$, $\mathbf{F} = \begin{pmatrix}\mathbf{I}_n & \mathbf{0}_{n,1} & \mathbf{0}_{n,1} \\ \mathbf{0}_{1,n} & 0 & 1 \\ \mathbf{0}_{1,n} & 0 & 0\end{pmatrix}$, $\mathbf{v}'_i = (\mathbf{v}_i^\top, -\gamma u_i, u_i)^\top$ and $h(\mathbf{x}_i, t_i) = h(\boldsymbol{\theta}_i, t_i) + n_i$. This formulation leads to the following assumption on the evolution noise $\mathbf{v}'_i$:

Assumption 2 *The evolution noise $\mathbf{v}'_i$ is additive and white with Covariance $\mathbf{P}_{\mathbf{v}'_i} = \mathbb{E}[\mathbf{v}'_i\mathbf{v}'^\top_i]$.*

3. Robust Kalman Temporal Difference

In this section we describe the Robust-KTD algorithm. This algorithm extends the XKTD algorithm (Geist and Pietquin, 2010) described in Section 2.2 by using the robust Bellman equation as the observation function, replacing Equation (2) by

$$h(\boldsymbol{\theta}_i, t_i) = \begin{cases}\hat{V}_{\boldsymbol{\theta}_i}(s_i) - \gamma \min_{p\in P} \sum_{s'\in\mathcal{S}} p(s'|s_i, a_i)\hat{V}_{\boldsymbol{\theta}_i}(s') \\ \hat{Q}_{\boldsymbol{\theta}_i}(s_i, a_i) - \gamma \min_{p\in P} \sum_{s'\in\mathcal{S}} \sum_{a'\in\mathcal{A}} p(s'|s_i, a_i)\hat{Q}_{\boldsymbol{\theta}_i}(s', a')\end{cases}, \tag{3}$$

where $t_i = (s_i, a_i)$. The *non-linear* function $h(\boldsymbol{\theta}_i, t_i)$ can be viewed as a variation of the Bellman equations presented in (1). The Robust-KTD algorithm receives as input the priors of the parameter vector $\hat{\boldsymbol{\theta}}_{0|0}$, its covariance matrix $\mathbf{P}_{0|0}$, the covariance of the evolution noise P_{v_i}, the variance of the white random process u_i, $\sigma^2_{n_i}$, and the parameters γ and κ. It initializes the augmented parameter vector to $\hat{\mathbf{x}}_{0|0} = \begin{pmatrix}\hat{\boldsymbol{\theta}}_{0|0}^\top & 0 & 0\end{pmatrix}$. At each time step i, the algorithm observes s_i, a_i, and r_i. Then it follows three steps: First, in the prediction step it updates the estimation of the augmented parameter vector, $\hat{\mathbf{x}}_{i|i-1} = \mathbf{F}\hat{\mathbf{x}}_{i-1|i-1}$ and the estimation of its covariance, $\mathbf{P}_{i|i-1} = \mathbf{F}\mathbf{P}_{i-1|i-1}\mathbf{F}^\top + \mathbf{P}_{v'_i}$.

In the second step the algorithm computes some statistics of interest. Since the observation function $h(\boldsymbol{\theta}_i, t_i)$ is non-linear these statistics are computed using the Unscented Kalman Filter (UKF) approach (Julier and Uhlmann, 1997). In this approach the augmented parameter distribution is represented by a minimal set of sample points, called *sigma-points*, that completely capture its mean and covariance. The algorithm collects $2(n+2)+1$ sigma points ($n+2$ is the dimension of $\hat{\mathbf{x}}_{\mathbf{i|i-1}}$) according to the following procedure: $\hat{\mathbf{x}}^{(0)}_{i|i-1} = \hat{\mathbf{x}}_{i|i-1}$, $\hat{\mathbf{x}}^{(j)}_{i|i-1} = \hat{\mathbf{x}}_{i|i-1} + \left(\sqrt{(n+2+\kappa)\mathbf{P}_{i|i-1}}\right)_j$ for $1 \le j \le n+2$ and $\hat{\mathbf{x}}^{(j)}_{i|i-1} = \hat{\mathbf{x}}_{i|i-1} - \left(\sqrt{(n+2+\kappa)\mathbf{P}_{i|i-1}}\right)_{n-j}$ for $n+3 \le j \le 2n+4$, where κ is a scaling factor that controls the accuracy and $(\sqrt{\mathbf{P}_X})_j$ is the j^{th} column of the Cholesky decomposition of P_X. The associated weights are $w_0 = \frac{\kappa}{n+2+\kappa}$ and $w_j = \frac{1}{2(n+2+\kappa)}$ for $1 \le j \le 2n+4$. These sample points are then propagated through the non-linear system: $\hat{r}^{(j)}_{i|i-1} = h(\hat{\boldsymbol{\theta}}_{i|i-1}, t_i) + \hat{n}^{(j)}_{i|i-1}$ and capture its posterior mean, $\hat{r}_{i|i-1} = \sum_{j=0}^{2n+4} \omega_j \hat{r}^{(j)}_{i|i-1}$ and its covariance, $P_{\hat{r}_{i|i-1}} = \sum_{j=0}^{2n+4} \omega_j (\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})^2$. The covariance of the augmented parameter vector and the innovation is computed by $\mathbf{P}_{\mathbf{x}r_i} = \sum_{j=0}^{2n+4} \omega_j (\hat{\mathbf{x}}^{(j)}_{i|i-1} - \hat{\mathbf{x}}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$.

The third step of the algorithm is the correction step where the algorithm computes the Kalman gain, $\mathbf{K}_i = \mathbf{P}_{\mathbf{x}r_i} P^{-1}_{\hat{r}_{i|i-1}}$, corrects the predicted augmented parameter vector,

$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + \mathbf{K}_i(r_i - \hat{r}_{i|i-1})$ and corrects the estimation of its covariance matrix, $\mathbf{P}_{i|i} = \mathbf{P}_{i|i-1} - \mathbf{K}_i P_{\hat{r}_{i|i-1}} \mathbf{K}_i^\top$. This procedure is repeated with each new observation. After k observations, the algorithm can be terminated and then it outputs the estimated parameter vector: $\hat{\boldsymbol{\theta}}_{k|k} = \hat{\mathbf{x}}_{k|k}^{(1:n)}$. We will show in Section 3.1 that the parameter estimations generated by the algorithm, $\hat{\boldsymbol{\theta}}_{i|i}$, minimize a regularized objective function.

The Robust-KTD algorithm involves a solution to the following inner problem at each time step: $\min_{p \in \mathcal{P}(s,a)} \sum_{s' \in \mathcal{S}(s,a)} p(s'|s) \hat{V}_{\hat{\boldsymbol{\theta}}}(s')$ where $\mathcal{S}(s,a)$ is the set of all possible next states from state s when taking action a under all transition probabilities in the set $\mathcal{P}(s,a)$. The solution to this problem may be computationally demanding when the set $\mathcal{P}(s,a)$ is large. Fortunately, there are some families of sets for which the solution is tractable, see for examples Iyengar (2005) and Nilim and El Ghaoui (2005).

We note that in our algorithm the transition probability p may be any $p \in \mathcal{P}(s,a)$ that satisfies the Rectangularity property (Iyengar, 2005). This property implies that at each time step, the selection of $\mathcal{P}$ for a certain state-action pair (s,a) is assumed to be independent of the actions chosen in other states and also of previously visited states and actions. The Robus-KTD algorithm does not require additional assumptions for the uncertainty set. This is a major advantage comparing to the RPVI algorithm, where the convergence analysis in Tamar et al. (2014) is based on the restrictive assumption over the transitions of the exploration policy and the (uncertain) transitions of the policy under evaluation.

3.1 Convergence Analysis

First, we adopt Assumptions 1 and 2 described in Section 2.2 for the models of the noises. Assumption 3 is concerned with the posterior distribution and the likelihood of the augmented parameter vector:

Assumption 3 *The posterior distribution $p(\mathbf{x}_i|r_{1:i-1})$ and the likelihood $p(r_i|\mathbf{x}_i)$ are assumed to be Gaussian with the following mean and covariance: $\mathbf{x}_i|r_{1:i-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{i|i-1}, \mathbf{P}_{i|i-1})$ and $r_i|\mathbf{x}_i \sim \mathcal{N}(h(\mathbf{x}_i, t_i), P_{n_i})$. In addition, we assume the following conditional independence: $p(r_i|r_{1:i-1}, \mathbf{x}_i) = p(r_i|\mathbf{x}_i)$.*

Theorem 1 *Under Assumptions 1, 2 and 3 the Robust-KTD algorithm minimizes at each time step i the following regularized cost function: $J(\mathbf{x}_i) = \frac{1}{2}\big(r_i - h(\mathbf{x}_i, t_i)\big)^\top P_{n_i}^{-1}\big(r_i - h(\mathbf{x}_i, t_i)\big) + \frac{1}{2}(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})^\top \mathbf{P}_{i|i-1}^{-1}(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})$, where $\hat{\mathbf{x}}_{i|i} \in \arg\min_{\mathbf{x}_i} J(\mathbf{x}_i)$ and the optimal parameter estimation is $\hat{\boldsymbol{\theta}}_{\boldsymbol{i|i}} = \hat{\mathbf{x}}_{i|i}^{(1:n)}$.*

The proof for Theorem 1 is based on the developments made for Sigma-Point Kalman Filters (SPKF) described in Van Der Merwe (2004). The objective function in Theorem 1 is driven from maximizing the log of the posterior distribution $p(\mathbf{x}_i|r_{1:i})$, under the Gaussian assumptions described in Assumption 3. The convergence is achieved due to the stochastic evolution model of the parameter vector, which treats the value function as a random variable and allows it to adjust to the colored observation noise.

4. Experiments

In this section we will examine the Robust-KTD algorithm in the Boyan chain domain. This experiment emphasis the advantage of our algorithm in estimating *on-line* the value function in RMDPs. We will compare the performance of our algorithm to the XKTD algorithm (Geist and Pietquin, 2010) and to the RPVI algorithm presented in Tamar et al. (2014).

The experiment procedure and the parameters we used are the same as in Geist and Pietquin (2010). The experiments proceeded as follows. First, we generated 50 trajectories with next state probability $p = 1$ for states $s \in \{0, 1\}$ and $p = 0.5$ for states $s \in \{2, \ldots, 13\}$. From these trajectories we estimated $\hat{p}_s$ for each state using the maximum likelihood estimator for p, and calculated the 95% confidence interval $[\hat{p}_{s-}, \hat{p}_{s+}]$ using the Clopper-Pearson method (Clopper and Pearson, 1934). This confidence interval was used to build the uncertainty set $\mathcal{P}(s)$ for each state. Then we simulated $N = 15$ trajectories using the estimated $\hat{p}_s$ for each state which we used as the input for the algorithms. The algorithms passed the same set of N trajectories in each iteration for a total of 20 iterations (due to this requirement in the RPVI algorithm).

The average and standard deviation results from 100 experiments are presented in the left plot in Figure 1. We used the error measure $\frac{1}{\|\boldsymbol{\theta}^*\|}\|\theta - \theta^*\|$ where $\boldsymbol{\theta}^*$ is the optimal parameter vector. In the top plot we compared the converged $\hat{\boldsymbol{\theta}}_i$ of each algorithm to the nominal optimal parameter $\boldsymbol{\theta}^* = [-24, -16, -8, 0]^\top$ which could be calculated accurately since the optimal value function is exactly linear in the features. As expected, XKTD achieves the lowest error since it provides an optimal solution when the state transitions are known or estimated correctly. In the bottom plot we used the robust optimal parameter which was calculated by the Robust-VI algorithm (RVI; Iyengar, 2005). We can see that Robust-KTD and RPVI converge to the same solution, however RPVI needs approximately 11 iterations to converge while Robust-KTD converged within the first iteration. This result emphasis the advantage of our algorithm in providing value function estimations on-line, as soon as an immediate reward is achieved. The XKTD algorithm converged to a biased result since it does not account for the uncertainties in the probabilities estimations. In the right plot in Figure 1 we present the errors from the first 100 samples within the first iteration. The fast convergence is typical to Kalman based algorithms.

5. Conclusions and Future Work

We presented an *on-line* method for solving both small and large-scale uncertain MDPs. This method supports linear and non-linear approximations for estimating the value function of robust policies. We derived the Robust-KTD algorithm, proved its convergence and examined its performance on a small scale domain. Future work should address accounting for changes in the confidence level during the evaluation procedure, according to historical data. Additional non-linear estimation procedures for RMDPs should also be considered within the Kalman filtering scheme.

Acknowledgments

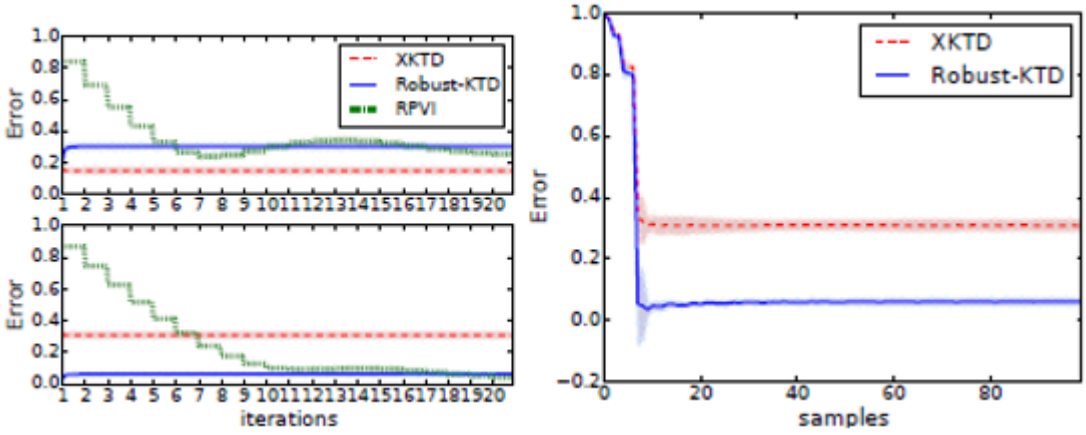

Figure 1: Convergence of XKTD, Robust-KTD and RPVI algorithms. **Left Top**: compared to the nominal optimal parameter; **Left Bottom**: compared to the robust optimal parameter; **Right**: Zoom in on the first iteration, compared to the robust optimal parameter

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Program (FP/2007-2013) / ERC Grant Agreement n. 306638.

References

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. In *Advances in Neural Information Processing Systems*, pages 1522–1530, 2015.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 201–208. ACM, 2005.

Matthieu Geist and Olivier Pietquin. Kalman temporal differences. *Journal of artificial intelligence research*, 39:483–532, 2010.

Ronald A Howard. Dynamic programming and Markov processes.. 1960.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30 (2):257–280, 2005.

Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

Vijaymohan R Konda and Vivek S Borkar. Actor-critic–type learning algorithms for Markov decision processes. *SIAM Journal on control and Optimization*, 38(1):94–123, 1999.

Shie Mannor, Duncan Simester, Peng Sun, and John N Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Marek Petrik and Shlomo Zilberstein. Robust approximate bilinear programming for value function approximation. *Journal of Machine Learning Research*, 12(Oct):3027–3063, 2011.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Aviv Tamar, Shie Mannor, Huan Xu, and EDU SG. Scaling up robust mdps using function approximation. In *ICML*, volume 32, page 2014, 2014.

Rudolph Van Der Merwe. *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. PhD thesis, Oregon Health & Science University, 2004.