Kalman Temporal Differences: Uncertainty and Value Function Approximation

Matthieu Geist[*], **Gabriel Fricout**
MCE Department, ArcelorMittal Research
Maizières-lès-Metz, FRANCE
matthieu.geist@supelec.fr

Olivier Pietquin
IMS Research Group, Supélec
Metz, FRANCE
olivier.pietquin@supelec.fr

Abstract

This paper deals with value (and Q-) function approximation in deterministic Markovian decision processes (MDPs). A general statistical framework based on the Kalman filtering paradigm is introduced. Its principle is to adopt a parametric representation of the value function, to model the associated parameter vector as a random variable and to minimize the mean-squared error of the parameters conditioned on past observed transitions. From this general framework, which will be called *Kalman Temporal Differences* (KTD), and using an approximation scheme called the unscented transform, a family of algorithms is derived. Contrary to most of function approximation schemes, this framework inherently allows to derive uncertainty information over the value function, which can be notably useful for the exploration/exploitation dilemma.

1 Introduction

Many approaches have been designed to handle the well known dilemma between exploration and exploitation, *e.g.* [3, 10, 14]. Uncertainty evaluation is a key part in handling this problem. Uncertainty can be over models or directly over values of given states, however this information is very useful. A problem which received little attention is how to compute value function uncertainty in the context of generalization. Sometimes L_∞ or L_2 bounds are given, however they are global and not local. To the best of our knowledge, the only approach able to provide uncertainty information about value function in such a context is based on Gaussian processes [4]. An equivalence between kernel ridge regression and Gaussian process regression is used in [9] to derive a similar uncertainty. This paper introduces a new function approximation scheme based on Kalman filtering which allows to derive uncertainty information at any point of the approximate function.

The focus is here on deterministic MDP $\{S, A, T, R, \gamma\}$, where S is the state space, A the action space, T the deterministic transition function, R the bounded reward function, and γ the discount factor. A policy π is a (here deterministic) mapping from states to actions. The value function of a given policy is classically defined as $V^\pi(s) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi]$ where r_i is the reward observed at time i, and similarly $Q^\pi(s,a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a, \pi]$. Reinforcement Learning (RL) [15] aims at finding the policy π^* which maximises the value function for every state: $\pi^* = \operatorname{argmax}_\pi(V^\pi)$. Two schemes (among others) can lead to the solution. First, policy iteration implies to learn the value function of a given policy, and then improve the policy, the new one being greedy respectively to the learned value function. It implies to solve the *Bellman evaluation equation*, which is given here for the value function and the Q-function, respectively:

$$V^\pi(s) = R(s, \pi(s), s') + \gamma V^\pi(s'), \forall s \tag{1}$$

$$Q^\pi(s,a) = R(s, a, s') + \gamma Q^\pi(s', \pi(s')), \forall s, a \tag{2}$$

[*]Matthieu Geist is also with Supélec and with CORIDA team, INRIA Lorraine, France.

Here and through the rest of the paper, s' denotes the transiting state, that is $s' = T(s, \pi(s))$ or $s' = T(s, a)$, depending on the context. The second scheme, value iteration, aims directly at finding the optimal policy without increment in the policy space. It implies to solve the *Bellman optimality equation*

$$Q^*(s,a) = R(s,a,s') + \gamma \max_{b \in A} Q^*(s',b),\ \forall s, a \tag{3}$$

The aim of this paper is to find an approximate solution of the Bellman evaluation or optimality equations, for the value function or the Q-function, when the state or action spaces are too large for classical approaches to hold. Moreover the proposed algorithm should be online, which is a RL property which should be kept, and above all it should allow to derive an uncertainty information for the value (or Q-) function for any state or action.

Temporal differences (TD) algorithms are a class of methods which consist in correcting the representation of the value (or Q-) function according to the TD error made on the value (or Q-) function. Most of them can be generically written as $\theta_{i+1} = \theta_i + K_i \delta_i$. In this expression, θ_i is the current representation of the value function, θ_{i+1} is an updated representation given an observed transition, δ_i is the so-called TD error, and K_i is a gain which indicates in which direction the representation of the value function should be corrected. Each of these terms is now discussed.

If the state space S and the action space A are finite and small enough, an exact description of the value function is possible, and θ is a tabular representation. If these spaces are too large, an approximation should be chosen. A classical choice in RL is the linear parameterization. Many function approximation algorithms require such a representation to ensure convergence [12], or even to be applicable [2]. Other representations are possible such as neural networks where θ contains the set of associated weights. Indeed, the proposed KTD framework is applicable to any representation of the value (or Q-) function, as long as it can be fully described by a finite set of p parameters.

The term δ_i is the TD error. Suppose that at time i a transition (s_i, a_i, s_{i+1}, r_i) is observed. For TD-like algorithms, that is algorithms which aim at evaluating the value function of a given policy π, the TD error is of the form $\delta_i = r_i + \gamma \hat{V}_{\theta_i}(s_{i+1}) - \hat{V}_{\theta_i}(s_i)$. For SARSA-like algorithms, that is algorithms which aim at evaluating the Q-function of a given policy π, the TD error is of the form $\delta_i = r_i + \gamma \hat{Q}_{\theta_i}(s_{i+1}, a_{i+1}) - \hat{Q}_{\theta_i}(s_i, a_i)$. Finally, for Q-learning-like algorithms, that is algorithms which aim at computing the optimal Q-function, the TD error is of the form $\delta_i = r_i + \gamma \max_{b \in A} \hat{Q}_{\theta_i}(s_{i+1}, b) - \hat{Q}_{\theta_i}(s_i, a_i)$. The type of TD error which is used determines which Bellman equation is to be solved, and thus if the algorithm is of the type policy or value iteration.

The term K_i is a gain. The most famous common are reviewed here. For TD, SARSA and Q-learning (see [15] for example), the gain can be written as $K_i = \alpha_i \sum_{j=1}^{i} \lambda^{i-j} e_j$ where α_i is a classical learning rate in stochastic approximation theory, and should verify $\sum_{i=0}^{\infty} \alpha_i = \infty$ and $\sum_{i=0}^{\infty} \alpha_i^2 < \infty$, e_j is an unitary vector which is zero everywhere except in the component corresponding to the state s_j (or to the state-action (s_j, a_j)) which is equal to one, and λ is the eligibility factor. These algorithms have also been extended to take into account approximate representations of the value function. According to [7] they are called direct algorithms. The gain can be written as $K_i = \alpha_i \sum_{j=1}^{i} \lambda^{i-j} \nabla_{\theta_i} \hat{V}_{\theta_i}(s_j)$ where $\nabla_{\theta_i} \hat{V}_{\theta_i}(s_j)$ is the derivation following the parameter vector of the parameterized value function in the state s_j. Note that the value function can be replaced straightforwardly by the Q-function in this gain. Another well known approach is the set of residual algorithms [7], for which the gain is obtained through the minimization of the L_2-norm of the Bellman residual: $K_i = \alpha_i \nabla_{\theta_i} (\hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}))$. The last approach we review is the Least-Squares Temporal Differences (LSTD) algorithm [2], which is only defined for linear parameterization and for which the gain is defined recursively.

So, a question holds, given a representation of the value function (or of the Q-function) and given a temporal differences scheme, what is the best gain K ? To answer this question, a statistical point of view is adopted here and the Kalman filtering framework [11] is followed.

2 Kalman Temporal Differences

In this section a very general point of view is adopted, and practical algorithms will be derived later. For now, a transition is generically noted as t_i and the shortcut g_{t_i} is adopted according to the

following notations:

$$t_i = \begin{cases} (s_i, s_{i+1}) \\ (s_i, a_i, s_{i+1}, a_{i+1}) \\ (s_i, a_i, s_{i+1}) \end{cases} \quad \text{and} \quad g_{t_i}(\theta_i) = \begin{cases} \hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \hat{Q}_{\theta_i}(s_{i+1}, a_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \max_{b \in A} \hat{Q}_{\theta_i}(s_{i+1}, b) \end{cases} \tag{4}$$

given that the aim is the value function evaluation (1), the Q-function evaluation (2) or the Q-function optimization (3). Thus all TD schemes can be written generically as $\delta_i = r_i - g_{t_i}(\theta_i)$.

As said before, a statistical point of view is adopted. The parameter vector θ_i is modeled as a random variable following a random walk. The problem at sight can thus be stated in a so-called *state-space formulation*:

$$\begin{cases} \theta_{i+1} = \theta_i + v_i \\ r_i = g_{t_i}(\theta_i) + n_i \end{cases} \tag{5}$$

The first equation is the evolution equation, it specifies that the parameter vector follows a random walk which expectation corresponds (approximately) to the optimal value function. The evolution noise v_i is centered, white and independent. The second equation is the observation equation, it links the observed transition to the value (or Q-) function through a Bellman equation. The observation noise n_i is supposed centered, white and independent. Notice that this necessary assumption does not hold for stochastic MDPs, that is why deterministic transitions are considered here. This model noise arises from the fact that the solution of the Bellman equation does not necessarily exists in the functional space spanned by the set of parameter vectors.

The objective could be to estimate the parameter vector which minimizes the expectation of the mean square error conditioned on past observed transitions. The associated cost can be written as:

$$J(\hat{\theta}_i) = E\left[\|\theta_i - \hat{\theta}_i\|^2 | r_{1:i}\right] \text{ with } r_{1:i} = r_1, r_2, \ldots, r_i \tag{6}$$

Generally speaking, the minimum mean square error (MMSE) estimator is the conditional expectation: $\operatorname{argmin}_{\hat{\theta}_i} J(\hat{\theta}_i) = \hat{\theta}_{i|i} = E\left[\theta_i | r_{1:i}\right]$. However, except in specific cases, this estimator is not computable. Instead, the aim here is to find the best *linear* estimator:

$$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i \tilde{r}_i \tag{7}$$

In equation (7), $\hat{\theta}_{i|i}$ is the estimate at time i, $\hat{\theta}_{i|i-1} = E[\theta_i | r_{1:i-1}]$ is the prediction of this estimate according to past rewards $r_{1:i-1}$, and for a random walk model the prediction is the previous estimation: $\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$. The innovation

$$\tilde{r}_i = r_i - \hat{r}_{i|i-1} \tag{8}$$

is the difference between the observed reward r_i and its prediction based on the previous estimate of the parameter vector:

$$\hat{r}_{i|i-1} = E\left[g_{t_i}(\theta_i) | r_{1:i-1}\right] \tag{9}$$

Using classical equalities, the cost function can be rewritten as:

$$\begin{aligned} J(\hat{\theta}_i) &= E\left[\|\theta_i - \hat{\theta}_i\|^2 | r_{1:i}\right] = E\left[(\theta_i - \hat{\theta}_i)^T (\theta_i - \hat{\theta}_i) | r_{1:i}\right] \\ &= \operatorname{trace}\left(E\left[(\theta_i - \hat{\theta}_i)(\theta_i - \hat{\theta}_i)^T | r_{1:i}\right]\right) = \operatorname{trace}\left(\operatorname{cov}\left(\theta_i - \hat{\theta}_i | r_{1:i}\right)\right) \end{aligned} \tag{10}$$

A first step is so to express the conditioned covariance over parameters as a function of the gain K_i. A few more notations are first introduced (recall also the definition of the innovation (8)):

$$\begin{cases} \tilde{\theta}_{i|i} = \theta_i - \hat{\theta}_{i|i} & \text{and} \quad \tilde{\theta}_{i|i-1} = \theta_i - \hat{\theta}_{i|i-1} \\ P_{i|i} = \operatorname{cov}\left(\tilde{\theta}_{i|i} | r_{1:i}\right) & \text{and} \quad P_{i|i-1} = \operatorname{cov}\left(\tilde{\theta}_{i|i-1} | r_{1:i-1}\right) \\ P_{r_i} = \operatorname{cov}\left(\tilde{r}_i | r_{i|i-1}\right) & \text{and} \quad P_{\theta r_i} = E\left[\tilde{\theta}_{i|i-1} \tilde{r}_i | r_{1|i-1}\right] \end{cases} \tag{11}$$

Using the postulated update of equation (7), and the various estimators being unbiased, the covariance can be expanded:

$$\begin{aligned} P_{i|i} &= \operatorname{cov}\left(\theta_i - \hat{\theta}_{i|i} | r_{1:i}\right) = \operatorname{cov}\left(\theta_i - \left(\hat{\theta}_{i|i-1} + K_i \tilde{r}_i\right) | r_{1:i-1}\right) \\ &= \operatorname{cov}\left(\tilde{\theta}_{i|i-1} - K_i \tilde{r}_i | r_{1:i-1}\right) = P_{i|i-1} - P_{\theta r_i} K_i^T - K_i P_{\theta r_i}^T + K_i P_{r_i} K_i^T \end{aligned} \tag{12}$$

The optimal gain can thus be obtained by deriving the trace of this matrix. First note that the gradient being linear, for three matrixes of *ad hoc* dimensions A, B and C, B being symmetric, the following algebraic identities hold: $\nabla_A \left(\text{trace}\left(ABA^T\right)\right) = 2AB$ and $\nabla_A \left(\text{trace}\left(AC^T\right)\right) = \nabla_A \left(\text{trace}\left(CA^T\right)\right) = C$, and thus using also equation (12):

$$\nabla_{K_i} \left(\text{trace}\left(P_{i|i}\right)\right) = 0 \Leftrightarrow \quad 2K_i P_{r_i} - 2P_{\theta r_i} = 0 \Leftrightarrow \quad K_i = P_{\theta r_i} P_{r_i}^{-1} \tag{13}$$

Using this optimal gain, the error covariance matrix is $P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$. Notice that no Gaussian assumption has been used to derive this algorithm.

The most general KTD algorithm, which breaks down in three stages, can now be derived. The first step consists in computing predicted quantities $\hat{\theta}_{i|i-1}$ and $P_{i|i-1}$. Recall that for a random walk model, the prediction is the previous estimation, and the predicted covariance can also be computed analytically:

$$P_{i|i-1} = \text{cov}\left(\tilde{\theta}_{i|i-1}|r_{1:i-1}\right) = \text{cov}\left(\tilde{\theta}_{i-1|i-1} + v_{i-1}|r_{1:i-1}\right) = P_{i-1|i-1} + P_{v_{i-1}} \tag{14}$$

where $P_{v_{i-1}}$ is the variance matrix of the evolution noise (which is known).

The second step is to compute some statistics of interest. It will be specialized later. The first statistic to compute is the prediction $\hat{r}_{i|i-1}$ (9). The second statistic to compute is the covariance $P_{\theta r_i}$ between the parameter vector and the innovation (11). However, from the state-space model (5), $r_i = g_{t_i}(\theta_i) + n_i$, and the observation noise is centered and independent, so

$$P_{\theta r_i} = E\left[\left(\theta_i - \hat{\theta}_{i|i-1}\right)\left(g_{t_i}\left(\theta_i\right) - \hat{r}_{i|i-1}\right)|r_{1:i-1}\right] \tag{15}$$

The last statistic to compute is the covariance of the innovation (11), which can be written (using again the characteristics of the observation noise):

$$P_{r_i} = E\left[\left(g_{t_i}(\theta_i) - \hat{r}_{i|i-1}\right)^2 |r_{1:i-1}\right] + P_{n_i} \tag{16}$$

where P_{n_i} is the variance of the observation noise.

The last step of the algorithm is the correction step. It consists in computing the gain (13), correcting the parameter vector (7) and correcting the associated covariance (12) accordingly. Notice that the matrix $P_{i|i-1}$ is the predicted error made on parameter estimate; it is used to compute the statistics of interest and it is a necessary quantity to compute (predicted) uncertainty over value (or Q-) function. Note also that as the proposed method is recursive, it must be initialized with some prioris $\hat{\theta}_{0|0}$ and $P_{0|0}$. The proposed general framework is summarized in algorithm 1. The main difficulty in applying KTD is to compute the statistics of interest $\hat{r}_{i|i-1}$, $P_{\theta r_i}$ and P_{r_i}, which is the subject of the next section.

3 Specializations

Analytic solutions to equations (9,15,16) can be derived for value function evaluation with linear parameterization. However the focus is here on more general cases involving nonlinearities. Moreover the Bellman optimality equation is considered in this framework. It implies to handle the $\max$ operator, which is non-derivable. As a consequence, local linearization is not sufficient. Computing statistics of interest can be state as computing first and second order moments of a nonlinearly mapped random variable. A useful approximation scheme, the unscented transform, is first introduced. It is then used to derive a set of three algorithms. Finally it is used to get uncertainty information of the value (or Q-) function for any given state.

3.1 The Unscented Transform

Let's abstract a little bit from RL and Kalman filtering. Let X be a random vector, and let Y be a mapping of X. The problem is to compute mean and covariance of Y knowing the mapping and first and second order moments of X. If the mapping is linear, classical analytical solution holds. If the mapping is nonlinear, the relation between X and Y can be generically written as $X = f(Y)$. A first solution would be to approximate the nonlinear mapping, that is to linearize

Algorithm 1: General KTD algorithm

Initialization: priors $\hat{\theta}_{0|0}$ and $P_{0|0}$;

for $i \leftarrow 1, 2, \ldots$ **do**

Observe transition t_i and reward r_i ;

Prediction step;
$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;
$P_{i|i-1} = P_{i-1|i-1} + P_{v_{i-1}}$;

Compute statistics of interest (using notably $\hat{\theta}_{i|i-1}$ and $P_{i|i-1}$);
$\hat{r}_{i|i-1} = E[g_{t_i}(\theta_i)|r_{1:i-1}]$;
$P_{\theta r_i} = E\left[(\theta_i - \hat{\theta}_{i|i-1})(g_{t_i}(\theta_i) - \hat{r}_i)|r_{1:i-1}\right]$;
$P_{r_i} = E\left[(g_{t_i}(\theta_i) - \hat{r}_{i|i-1})^2|r_{1:i-1}\right] + P_{n_i}$;

Correction step;
$K_i = P_{\theta r_i} P_{r_i}^{-1}$;
$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i\left(r_i - \hat{r}_{i|i-1}\right)$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

it around the mean of the random vector X, which leads to $E[Y] \approx f(E[X])$ and $E[YY^T] \approx (\nabla f(E[X]))\, E[XX^T]\, (\nabla f(E[X]))^T$. This approach is the base of Extended Kalman Filtering (EKF) [13], which has been extensively studied and used in past decades. However it has some limitations. First it cannot handle non-derivable nonlinearities ($\max$ operator). It also supposes that the nonlinear mapping is locally linearizable, which is unfortunately not always the case and can lead to quite bad approximation, as exemplified in [8].

The basic idea of the unscented transform is that it is easier to approximate an arbitrary random vector than an arbitrary nonlinear function. Its principle is to sample *deterministically* a set of so-called sigma-points from the expectation and the covariance of X. The images of these points through the nonlinear mapping f are then computed, and they are used to approximate statistics of interest. It shares similarities with Monte-Carlo methods, however here the sampling is deterministic, nonetheless allowing a given accuracy [8].

The original unscented transform is now described more formally (some variants have been introduced since, but the basic principle is the same). Let n be the dimension of the random vector X. A set of $2n+1$ sigma-points is computed as follows:

$$\begin{cases} x_0 = \bar{X} & w_0 = \frac{\kappa}{n+\kappa}, \quad j = 0 \\ x_j = \bar{X} + \left(\sqrt{(n+\kappa)P_X}\right)_j & w_j = \frac{1}{2(n+\kappa)}, \quad 1 \leq j \leq n \\ x_j = \bar{X} - \left(\sqrt{(n+\kappa)P_X}\right)_{n-j} & w_j = \frac{1}{2(n+\kappa)}, \quad n+1 \leq j \leq 2n \end{cases} \tag{17}$$

where $\bar{X}$ is the mean of X, P_X is its variance matrix, κ is a scaling factor which controls the accuracy of the unscented transform [8], and $\left(\sqrt{(n+\kappa)P_X}\right)_j$ is the j^{th} column of the Cholesky decomposition of the matrix $(n+\kappa)P_X$. Then the image through the mapping f is computed for each of these sigma-points: $y_j = f(x_j)$, $0 \leq j \leq 2n$. The set of sigma-points and their images can then be used to compute first and second order moments of Y, and even P_{XY}, the covariance between X and Y:

$$\bar{Y} \approx \sum_{j=0}^{2n} w_j y_j,\ P_Y \approx \sum_{j=0}^{2n} w_j\,(y_j - \bar{y})\,(y_j - \bar{y})^T \text{ and } P_{XY} \approx \sum_{j=0}^{2n} w_j\,(x_j - \bar{x})\,(y_j - \bar{y})^T \tag{18}$$

where $\bar{x} = x_0 = \bar{X}$ and $\bar{y} = \sum_{j=0}^{2n} w_j y_j$. The unscented transform having been presented, the specialization of KTD is now addressed.

3.2 KTD-V, KTD-SARSA and KTD-Q

The unscented transform having been introduced, the general KTD framework is specialized to the value function evaluation (KTD-V), the Q-function evaluation (KTD-SARSA) and the Q-function optimization (KTD-Q). Recall that the problem in applying KTD is to compute the statistics of interest given in equations (9,15,16). The unscented transform allows to approximate these quantities.

The three algorithms share the same computation of sigma-points from known statistics $\hat{\theta}_{i|i-1}$ and $P_{i|i-1}$ as described in section 3.1, as well as associated weights:

$$\Theta_{i|i-1} = \left\{ \hat{\theta}^{(j)}_{i|i-1},\ 0 \leq j \leq 2p \right\} \quad \text{and} \quad \mathcal{W} = \{w_j,\ 0 \leq j \leq 2p\} \tag{19}$$

The images of sigma-points have then to be computed, according to observation equation of state-space model 5. This step is specialized for each algorithm:

$$\mathcal{R}_{i|i-1} = \begin{cases} \left\{ \hat{r}^{(j)}_{i|i-1} = \hat{V}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i) - \gamma \hat{V}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}),\ 0 \leq j \leq 2p \right\} \text{ (KTD-V)} \\ \left\{ \hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i) - \gamma \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a_{i+1}),\ 0 \leq j \leq 2p \right\} \text{ (KTD-SARSA)} \\ \left\{ \hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i) - \gamma \max_{b \in A} \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, b),\ 0 \leq j \leq 2p \right\} \text{ (KTD-Q)} \end{cases} \tag{20}$$

Then the sigma-points and their images can be used to compute the statistics of interest as exemplified in section 3.1. The associated equations are the same for the three algorithms:

$$\begin{cases} \hat{r}_{i|i-1} &= \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1} \\ P_{r_i} &= \sum_{j=0}^{2p} w_j \left(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1} \right)^2 + P_{n_i} \\ P_{\theta r_i} &= \sum_{j=0}^{2p} w_j \left(\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1} \right) \left(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1} \right) \end{cases} \tag{21}$$

This gives a practical implementation of the general KTD in the case of value function evaluation, Q-function evaluation and Q-function optimization, which is summarized in algorithm 2.

3.3 Computing Uncertainty on Value

Given the KTD algorithm, the first and second order moments of the parameter vector, which is modelled as a random variable, is available. However, being a mapping of the parameter vector, the value (or Q-) function is a random function. The issue addressed here is how to compute associated mean and variance. The solution is once again based on the unscented transform.

This section focuses on the value function, extension to Q-function is straightforward. Suppose that the value function $\hat{V}_\theta$ is parameterized by the random vector θ of associated mean $\hat{\theta}$ and covariance P_θ. The set of sigma points $\Theta = \{\theta^{(j)},\ 0 \leq j \leq 2p\}$ and associated weights $\mathcal{W} = \{w_j,\ 0 \leq j \leq 2p\}$ can be computed (see section 3.1). Then, to compute the estimation of the value for a given state s and the associated covariance the following set of images of sigma-points must be computed: $\mathcal{V}(s) = \left\{ \hat{V}^{(j)}(s) = \hat{V}_{\theta^{(j)}}(s), \quad 0 \leq j \leq 2p \right\}$. Given the images and the associated weights, the prediction $\hat{V}(s)$ and the associated variance $\hat{\sigma}^2_V(s)$ can be approximated by:

$$\hat{V}(s) = \sum_{j=0}^{2p} w_j \hat{V}^{(j)}(s) \quad \text{and} \quad \hat{\sigma}^2_V(s) = \sum_{j=0}^{2p} w_j \left(\hat{V}^{(j)}(s) - \hat{V}(s) \right)^2 \tag{22}$$

Thus, given a representation θ of the value (or Q-) function, it is quite easy to compute associated estimation and variance of the value (or Q-) function for any given state s. This is exemplified in [5] for a simple regression problem. So, at each time step, an estimate $\hat{\theta}_{i|i}$ and the associated matrix error $P_{i|i}$ are available, and the unscented transform is used to propagate the uncertainty from parameters (modelled as $P_{i|i}$) to values (for any state).

4 Discussion and Perspectives

A general Kalman-based function approximation scheme for RL in deterministic MDPs has been introduced, and algorithms for value function and Q-function evaluation (policy iteration scheme)

Algorithm 2: KTD-V, KTD-SARSA and KTD-Q

Initialization: priors $\hat{\theta}_{0|0}$ and $P_{0|0}$;

for $i \leftarrow 1, 2, \ldots$ **do**

Observe transition $t_i = \begin{cases} (s_i, s_{i+1}) \text{ (KTD-V)} \\ (s_i, a_i, s_{i+1}, a_{i+1}) \text{ (KTD-SARSA)} \\ (s_i, a_i, s_{i+1}) \text{ (KTD-Q)} \end{cases}$ and reward r_i ;

Prediction Step;

$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;

$P_{i|i-1} = P_{i-1|i-1} + P_{v_{i-1}}$;

Sigma-points computation ;

$\Theta_{i|i-1} = \left\{ \hat{\theta}^{(j)}_{i|i-1}, \quad 0 \leq j \leq 2p \right\}$ (from $\hat{\theta}_{i|i-1}$ and $P_{i|i-1}$);

$\mathcal{W} = \{ w_j, \quad 0 \leq j \leq 2p \quad \}$;

$\mathcal{R}_{i|i-1} = \begin{cases} \left\{ \hat{r}^{(j)}_{i|i-1} = \hat{V}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i) - \gamma \hat{V}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}),\ 0 \leq j \leq 2p \right\} \text{ (KTD-V)} \\ \left\{ \hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i) - \gamma \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a_{i+1}),\ 0 \leq j \leq 2p \right\} \text{ (KTD-SARSA)} \\ \left\{ \hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i) - \gamma \max_{b \in A} \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, b),\ 0 \leq j \leq 2p \right\} \text{ (KTD-Q)} \end{cases}$;

Compute statistics of interest;

$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1}$;

$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$;

$P_{r_i} = \sum_{j=0}^{2p} w_j \left(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1} \right)^2 + P_{n_i}$;

Correction step;

$K_i = P_{\theta r_i} P_{r_i}^{-1}$;

$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i \left(r_i - \hat{r}_{i|i-1} \right)$;

$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

and for Q-function direct optimization (value iteration scheme) have been derived from it, as well as a way to compute value uncertainty for any state. Experimental results are not given here, however the KTD-Q algorithm has been first introduced in [6] from a Bayesian perspective, and related experiments are provided. Experimental results are promising.

The proposed framework has some potential advantages. First it does not suppose stationarity. An immediate application is to handle non-stationary environments. But an even more interesting one is the control case. The algorithm LSTD [2] is known to fail when combined with optimistic policy iteration, because of the induced non-stationarities of this specific learning and control scheme. Kalman filtering and thus the proposed framework is designed to be robust to non-stationarities (random walk model of the parameter vector). This can be quite interesting for the control case, which has not been treated in this paper (the focus was on learning the value function or the Q-function and associated uncertainty, given observed transitions, and not on how to choose action for a given state). Second, the parameter vector is modelled as a random vector. As a consequence, at each time step, the covariance of this random vector is available. It can be propagated to the value function in order to provide uncertainty information for the value at a given state, as demonstrated in section 3.3. This uncertainty propagation can be useful to handle the well known dilemma between exploration and exploitation. For now this uncertainty can be computed, however how to use it is still an open research problem.

Yet the proposed framework presents a major drawback. In the case of stochastic transitions, the KTD can produce biased estimates of the parameters, or even be unstable. The problem lies in the fact that the KTD minimizes a squared Bellman residual (see [16] for the demonstration of the minimized cost function with unscented filtering and random walk evolution model):

$J_i(\theta) = \sum_{j=1}^{i} P_{n_j}^{-1}(r_j - g_{t_j}(\theta))^2$. The cost function which should be considered to truly minimize the squared Bellman residual is $L(\theta) = \|V_\theta - TV_\theta\|^2$ where T is one of the Bellman operators (depending on the function we want to solve and involving transition probabilities). As noted in [1], $J_i(\theta)$ is a biased estimator of $L(\theta)$, the bias being a variance term which favorises smooth value functions. The same problem arises in the residual approach of [7]. A solution could be to introduce an auxiliary filter, in the same manner an auxiliary function has been introduced in [1]. Another solution could be to adapt the colored noise model used in [4]. For now, KTD can be applied without modification to stochastic environments (see [6] for a successful application of KTD-Q to a stochastic problem), but it can become unstable, depending on the problem at sight.

To finish with, most interesting perspectives are to extend the framework to the control case, for which the non-stationarity hypothesis and the uncertainty propagation should be useful, and to handle more rigorously stochastic transitions. It is also planned to conduct more comparisons, theoretically and experimentally, of KTD to other related function approximation schemes.

References

[1] András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, April 2008.

[2] Steven J. Bradtke and Andrew G. Barto. Linear Least-Squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996.

[3] Richard Dearden, Nir Friedman, and Stuart J. Russell. Bayesian Q-Learning. In *AAAI/IAAI*, pages 761–768, 1998.

[4] Yaakov Engel. *Algorithms and Representations for Reinforcement Learning*. PhD thesis, Hebrew University, April 2005.

[5] Matthieu Geist, Olivier Pietquin, and Gabriel Fricout. A Sparse Nonlinear Bayesian Online Kernel Regression. In *Proceedings of the Second IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (AdvComp 2008)*, pages 199–204, Valencia (Spain), October 2008.

[6] Matthieu Geist, Olivier Pietquin, and Gabriel Fricout. Bayesian Reward Filtering. In S. Girgin et al., editor, *Proceedings of the European Workshop on Reinforcement Learning (EWRL 2008)*, volume 5323 of *Lecture Notes in Artificial Intelligence*, pages 96–109. Springer Verlag, Lille (France), June 2008.

[7] Leemon C. Baird III. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning (ICML 95)*, pages 30–37, 1995.

[8] Simon J. Julier and Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, March 2004.

[9] Tobias Jung and Daniel Polani. Kernelizing LSPE(λ). In *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 338–345, 2007.

[10] Sham Kakade, Michael J. Kearns, and John Langford. Exploration in metric state spaces. In *ICML*, pages 306–312, 2003.

[11] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[12] Ralph Schoknecht. Optimality of Reinforcement Learning Algorithms with Linear Function Approximation. In *Conference on Neural Information Processing Systems (NIPS 15)*, 2002.

[13] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley & Sons, 1. auflage edition, August 2006.

[14] Alexander L. Strehl and Michael L. Littman. An Analysis of Model-Based Interval Estimation for Markov Decision Processes. *Journal of Computer and System Sciences (accepted for publication)*, July 2006 (submission).

[15] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 3rd edition, March 1998.

[16] Rudolph van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University, Portland, OR, USA, April 2004.