The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

GAUSS-MARKOV-KALMAN REINFORCEMENT LEARNING FOR TEMPORAL DIFFERENCE USING AN ENSEMBLE

VASOS ARNAOUTIS[1] and BOJANA ROSIC

[1] University of Twente, Netherlands
e-mail: v.arnaoutis@utwente.nl

Key words: Reinforcement learning, Kalman filter, Stochastic Optimization, Conditional Expectation

Summary. Temporal-Difference reinforcement learning can be seen as an inverse problem in a probabilistic setting through Bayes's rule. In particular, we formulate a learning problem by approximating the posterior by the conditional expectation and variance. We show that the resulting Temporal-Difference method arises from the generalization of classical Kalman-based reinforcement learning. The corresponding stochastic formulation of Temporal-Difference is further discretized by an ensemble method. The method is tested on a 2D control problem of an escape room.

1 Introduction

Model-free reinforcement learning [1] has gained more prominence in control and optimization applications. Observing information within an unfamiliar environment can help formulate strategies in the decision-making process across control and design domains [2]. This approach is beneficial in the case of complicated systems because one can avoid physics-based modeling.

In the context of decision-making under expert knowledge, the Bayesian framework emerges as a notable paradigm [3]. The ability to account for prior knowledge and the expression of uncertainty over decision-making is a critical component that benefits reinforcement learning in explainability and exploration [4, 5]. Stochastic reinforcement learning aims to improve convergence and sampling efficiency by adjusting each sample's importance during training or selective exploration, which reduces the sampling effort [6, 7].

This work establishes the reinforcement learning theory for model-free value function approximation algorithms in a Bayesian framework [8, 9]. The approach is based on a temporal difference (TD) learning, defined through Bayesian conditional moments of the value function, adding existing experts knowledge (prior information) and addressing process and measurement noise [10].

Existing implementations show the possibility of a probabilistic framework in various contexts. In the field of model-free algorithms and, specifically, value function approximation algorithms, these can be already summarised by existing literature reviews [11, 12]. Prior research has demonstrated the application of Kalman filter to TD, as exemplified by the Kalman Temporal Difference (KTD) framework [4]. While the KTD theory is derived from the unscented Kalman filter theory, it cannot fully approximate nonlinear reward-to-state mapping due to affine representations in the update equation. A similar approach is the one of Gaussian Process

Temporal Difference (GPTD) learning [13, 14]. GPTD learning models the prior of the value function by using Gaussian processes for stochastic transitions and rewards. The algorithm is further improved by introducing a weighted ensemble of Gaussian priors to enhance the representation of function space over a single kernel prior [15]. Another expansion of temporal difference is successor representation, which models the value function through a successor representation matrix and the reward that generalizes over states that predict similar outcomes [16]. A multimodal optimization further improves this by multiple initializations of the priors and variable basis functions, which are additionally optimized by a gradient descent method [17]. A modified extended Kalman filter loss function compatible with non-linear function approximators such as neural networks has also been developed to deal with higher dimensionality systems [18].

While classical Kalman filter theory represents only a special case of Bayesian learning, in this paper we propose more general variant based on the notion of conditional expectation (CE). The main idea is to estimate the first two moments of the posterior directly without assuming linear approximations like in a classical setting [9, 19], and not restrict ourselves to Gaussian measure.

Conditional expectations (CE) are closely related to the Bayesian update and can be effectively expressed by functional approximation and sampling methods in stochastic problems [9]. The utility of CE is shown in the Gauss-Markov theorem and its extensions related to the Kalman filter, defined as a linear approximation of the conditional expectation [19, 9]. While known, this association has not been explicitly established in the case of temporal-difference reinforcement learning. This paper demonstrates the relation between the return and conditional expectations and shows a derivation of the temporal difference learning through the Gauss-Markov theorem. In the linear case and assuming Gaussian distributions, this turns out to be the Kalman filter update equation [20], and under the right assumptions for discretization and noises, it is identical to the KTD algorithm. The paper derives the linear case of GMKF-TD with a discretization of the value function using an ensemble filter.

2 Reinforcement learning

The decision-making process of reinforcement learning is based on the Markov decision process (MDP) described by the tuple $\{S, A, R, \gamma\}$ [21, 22], in which S and A are the state and action sets, respectively, R is the reward that can be observed for each transition and γ is the discount factor used to diminish the impact of rewards collected far in the future. An MDP can be sufficiently described by the current state and action taken, with conditional independence to previous historical states and actions. The objective of reinforcement learning is to maximize the return G_t, described by the discounted cumulative rewards over a trajectory of T steps, i.e.

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k. \tag{1}$$

To optimize policy π, the goal is to maximize the expected cumulative reward

$$V_\pi(s) = \mathbb{E}_\pi[G_t | S = s], \quad \forall s \in S \tag{2}$$

also known as the state value function. The expectation is given for a policy π that is followed for selecting future actions given the state s.

3 Gauss-Markov-Kalman Temporal Difference

Let the unknown future cumulative reward G_t be modeled as uncertain in $L_2(\Omega, \mathfrak{F}, \mathbb{P})$. In other words, we model G_t based on prior (expert) knowledge as a finite variance random variable (RV) $G_t(\omega)$ in a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$ in which Ω is the sample space, $\mathfrak{F}$ is a σ-algebra of measurable events, and $\mathbb{P}$ is a probability measure. Given states S one can define a sub-σ-algebra $\mathfrak{B}$ where $\mathfrak{B} = \sigma(S) \subset \mathfrak{F}$ is generated by S. Then, one may define the conditional expectation

$$\mathbb{E}(G_t|\mathfrak{B}) := P_{\mathfrak{B}}(G_t) = \arg\min_{\hat{G}_t} ||G_t - \hat{G}_t|| \tag{3}$$

as orthogonal projection of G_t on the space of RVs generating measurement [23]. This further leads to orthogonal decomposition of the cumulative reward:

$$G_t = P_{\mathfrak{B}} G_t + (I - P_{\mathfrak{B}}) G_t. \tag{4}$$

Having Doob-Dynkin lemma [24], one may further state

$$\mathbb{E}(G_t|\mathfrak{B}) = \phi(S) \tag{5}$$

in which $\phi(S)$ is a measurable function of state S. Substituting the previous equation into Eq 4, one can write

$$G_t = \phi(S) + (G_t - \phi(S)). \tag{6}$$

Instead of directly observing the states, one is often observing some indirect function of them. Hence, one may further write

$$\mathbb{E}(G_t|\mathfrak{B}_g) = \varphi(g(S)) \tag{7}$$

in which $\mathfrak{B}_g := \sigma(g(S))$, and g is an observation function of the state. Let $R := g(S)$ be a reward observed in a state S then

$$\mathbb{E}(G_t|\mathfrak{B}_g) = \varphi(R) \tag{8}$$

and Eq. (6) can be further written as

$$G_t = \varphi(R) + (G_t - \varphi(R)). \tag{9}$$

Given data for the reward R and the prior information $R(\omega)$, one may further write

$$G_t^a(\omega) = \varphi(R) + (G_t(\omega) - \varphi(R(\omega)). \tag{10}$$

This equation represents the assimilation process of data and prior knowledge, denoted by the superscript a. Here, $G_t(\omega)$ on the left side is our prior random variable, and $R(\omega)$ is a forecast of the reward. Averaging Eq. (10) with respect to policy π and taking specific state $S = s$, one may further write

$$\mathbb{E}_\pi(G_t(\omega)|S = s) = \mathbb{E}_\pi(\varphi(R)|S = s) + \mathbb{E}_\pi(G_t(\omega)|S = s) - \mathbb{E}_\pi(\varphi(R(\omega))|S = s). \tag{11}$$

As $V(s, \omega) := \mathbb{E}_\pi(G_t(\omega)|S = s)$ is a random variable describing our prior knowledge about the state value function, one may further rewrite previous equation as

$$V(s,\omega) = V(s,\omega) + \mathbb{E}_\pi(\varphi(R)|S=s) - \mathbb{E}_\pi(\varphi(R(\omega)|S=s). \tag{12}$$

The function $\varphi(\cdot)$ can have any form that expresses the possible non-linearity of the mapping. In the simplest case, one can assume a linear function (affine) such that $\varphi(R(s)) = KR(s)+b$, where K is the optimal Kalman gain and b is a bias term. The linear coefficients can be computed by minimizing the orthogonal residual [19, 8], i.e.,

$$(K,b) = \arg\min_{\mathcal{K},\mathcal{B}} \|G_t(\omega) - \mathcal{K}R(\omega) - \mathcal{B}(\omega)\|^2. \tag{13}$$

Following this, Eq 12 reads

$$V(\omega) = V(\omega) + K(r(s) - r(s,\omega)) \tag{14}$$

in which $r(s) = \mathbb{E}_\pi(R(s)|S=s)$ is the observed mean reward and $r(s,\omega)$ is the forecast of the mean reward. It should be clear now that the innovation $r(s) - r(s,\omega)$ is the so-called temporal difference error. From Eq. (2), one can derive a recursive form of the value function, namely the Bellman equation defined as

$$V(s,\omega) \quad = \quad \mathbb{E}_\pi(R(s,\omega) + \gamma G_{t+1}(\omega)|S=s) \tag{15}$$

which can be rewritten for the forecast reward as

$$r(s,\omega) \quad = \quad V(s,\omega) - \gamma V(s',\omega) + \epsilon_r \tag{16}$$

$$r(s,\omega) \quad = \quad \hat{r}(s,\omega) + \epsilon_r \tag{17}$$

in which ϵ_r denotes the modelling error, and s' is the next state. Substituting Eq. (16) and Eq. (17) back into Eq. (12) one has

$$V^a(s,\omega) = V(s,\omega) + K(r(s) - V(s,\omega) + \gamma V(s',\omega) - \epsilon_r). \tag{18}$$

Here, the gain K can be derived by minimizing the loss of the value function [19]. From Eq. 13 the gain is derived as

$$K = C_{V(s,\omega),\hat{r}(s,\omega)}(C_{\hat{r}(s,\omega)} + C_{\epsilon_r})^\dagger \tag{19}$$

where $C_{V(s,\omega),r(s,\omega)}$ is the covariance between the forecast reward and value function, and C_R, C_{ϵ_r} are the auto-covariance of the reward and modeling error. The above filter is called the Gauss-Markov-Kalman Filter TD (GMKF-TD) as it is an extract of the theory presented in [9], implemented for temporal difference. In the linear (affine) case, GMKF-TD is the same as KTD. From Eq. (15) and Eq. (16), it is clear that the forecast of the reward depends on the transitioned probability for state s'. This recursion results in a biased estimate of the value function [25]. One method to remove the bias is using a colored noise for ϵ_r [14]. In the simplest case, the modeling error ϵ_r can be modeled as white noise. However, this is shown to produce a biased value function [4].

3.1 Parameterization

Solving Eq. (18) requires parameterization of the value function. One may assume the following parameterization

$$V_s \approx V_s(\theta) = \theta^T \Phi(s) \tag{20}$$

with $\Phi(s)$ being a known kernel function and θ a vector of parameters. For the prior, the previous equation leads to

$$V_t(\omega) \approx V_{\theta_t}(s, \omega) = \theta_t^T(\omega)\Phi(s) \tag{21}$$

depending on its parameters and time such that, $\theta_t(\omega) : \Omega_\theta \times T \to \mathbb{R}^k$. For non-stationary MDP, or any other time series dependency, where parameter updates depend on time, one can account for changes in the structure of the value function apriori. One way to formulate this is using an additive process, assuming that $\eta_t(\omega)$ is independent of $\theta_{t-1}(\omega)$:

$$\theta_t(\omega) = \theta_{t-1}(\omega) + \eta_t(\omega). \tag{22}$$

Substituting Eq. (22) and Eq. (21) to Eq. (18) one obtains

$$\theta^a(\omega) \quad = \theta(\omega) + K(r(s) - V_\theta(s, \omega) + \gamma V_\theta(s', \omega) - \epsilon_r) \tag{23}$$

in which

$$K \quad = C_{\theta(\omega), r(s,\omega)}(C_{r(s,\omega)} + C_{\epsilon_r})^{-1} \tag{24}$$

4 Discretization - Ensemble filter

The previously derived rule of updating the value function is continuous w.r.t. the prior probability measure. Thus, a discretization is required. For this purpose we employ the Monte Carlo method. In other words, Eq. (22) is sampled such that

$$\theta^a(\omega_j) = \theta(\omega_j) + K(r(s) - V_\theta(s, \omega_j) + \gamma V_\theta(s', \omega_j) - \epsilon_r) \tag{25}$$

where ω_j indicates one sample out of the ensemble. The ensemble is denoted by $\Theta = [\theta(\omega_1), ..., \theta(\omega_J)]$. From the ensemble, the Kalman gain is computed by

$$K = C_{\Theta, r}(C_r + C_{\epsilon_r})^{-1} \tag{26}$$

where the covariances can be computed directly from the ensemble [26]. Given $\mathbf{1}_J^T$ is a row of vector of 1s of same length as the ensemble, the mean and its variance for the parameters Θ, and respectively $r_\Theta(s)$, can be computed by

$$\mathbb{E}(\theta) = \bar{\Theta} \quad = \quad \frac{1}{J} \sum_j \theta(\omega_j) \tag{27}$$

$$C_\theta = \mathbb{E}[\theta \otimes \theta] \quad = \quad \frac{1}{J-1} \mathbb{E}[(\Theta - \bar{\Theta}\mathbf{1}_J^T)(\Theta - \bar{\Theta}\mathbf{1}_J^T)^T]. \tag{28}$$

5 Mean Temporal Difference

By now, it should be clear that the formulation of GMKF-TD is an approximate Bayesian representation of classical temporal difference reinforcement learning. This can be proven directly from Eq. (18) by taking the expectation of $V(s)$, which would result in a scalar value i.e., the mean value that reads

$$\bar{V}^a(s) = \bar{V}(s) + k(R(s) - \bar{V}(s) + \gamma \bar{V}(s')) \tag{29}$$

where the gain k cannot be computed through the covariance anymore, assuming that we are dealing only with first order moments. In the simplest case and for linear parameterization, k can be set as a small value multiplied by the kernel basis function such that $k = \alpha_k \Phi(s)$, where $\alpha_k << 1$ is called the learning rate. Literature introduced a variety of heuristic approaches to set up the learning rate [27], however analysing this goes beyong the scope of the paper. For the implementation of the example, an adaptive learning rate is chosen where the learning rate is determined by

$$\alpha_k(t) = \alpha_k(0) \frac{j+1}{j+N} \tag{30}$$

where $\alpha_k(0)$ is the initial learning rate, N is the current iteration (often defined by episodes) and j is an arbitrary number.

6 Numerical Results

The implementation of the escape room is based on the "simple maze" environment presented by [4] to showcase the utilization of the algorithm. The escape room describes a 2D room of dimensions $(x, y) \in [0, 1]^2$ through which the agent is required to navigate. The behavioral policy of the agent is described through four actions (up, down, left, right) moving the agent 0.05 distance in the respective direction with probability of occurrence $P(\text{up}) = 0.9$ and $P(\text{down}) = P(\text{left}) = P(\text{right}) = 0.1/3$. The agent's position is initialized at position (x_0, y_0) with $x_0 \sim \mathcal{N}(\frac{1}{2}, \frac{1}{8})$ and $y_0 \sim \mathcal{U}(0, 0.05)$. The reward is constructed such that

$$R = \begin{cases} +1 & \text{if } y = 1 \text{ and } x \in [\frac{3}{8}, \frac{5}{8}] \\ -1 & \text{if } y = 1 \text{ and } x \in [[0, \frac{3}{8}] \cup [\frac{5}{8}, 1]] \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

The simulation is run indefinitely until either a reward of $+1$ or -1 is observed, after which it is terminated and reset. If the agent reaches the environment bounds, the new agent position is clipped within the bound dimensions.

The problem is further expressed through some expert knowledge to facilitate temporal difference training. For this, the space is discretized over a radial basis function kernel $\Phi(s)$ centered at points $\{0, 0.5, 1\} \times \{0, 0.5, 1\}$ and with standard deviation of 1. For Eq. (25), the model error is set to $C_{\epsilon_r} = 1$ and prior is assumed (wide sense) stationary such that $\theta_{i+1} = \theta_i$ (i.e, $\eta_t(\omega) = 0$ from Eq. (22)) . The discount factor was set to $\gamma = 0.9$.

The ensemble is constructed from 1000 samples. The value function approximation parameters are a matrix Θ in $\mathbb{R}^{n_i, n_j}$ where n_i are the number of features in the vector $\Phi(s)$ and n_j the number of samples. The ensemble prior is set at $\mathbb{E}[\theta] = \mathbf{0}$ and $\mathbb{E}[\theta \otimes \theta] = 10I$.

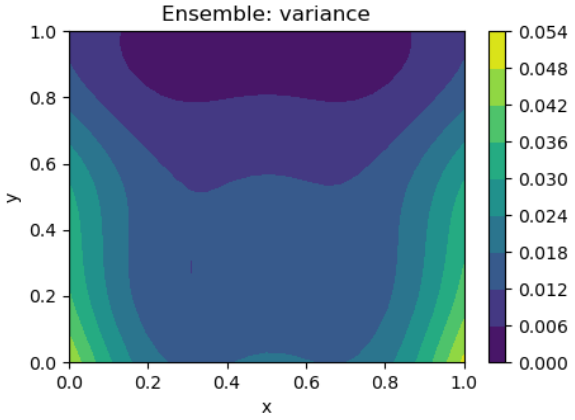

Figure 1: Plot of variance of value function for escape room environment using Ens-GMKF-TD

The variance of the value function for the escape room, computed by the ensemble GMKF-TD is shown in Figure 1, after 100 episodes. From this, one can see that the variance expresses confidence over the learned mean values, that approaches zero after each visitation.

7 Conclusion

A reformulation of reinforcement learning is presented in a Bayesian framework of probability density functions, based on conditional expectation. An affine representation of the reward function is shown where in the linear case is a formulation of Kalman filter temporal difference, namely GMKF-TD. In the mean sense, this is simplified to the classical temporal difference learning. A discretization choice is made for representing the RV of the value function through an ensemble, from which covariances can be computed directly. The algorithm implemented is in a toy example, for which the importance of computing higher moments is showcased.

REFERENCES

[1] M. Wiering, M. van Otterlo, Reinforcement Learning, Vol. 12, Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-27645-3.

[2] V. Arnaoutis, B. Rosic, E. Lutters, Automated design evolution for parametric design applied on computer-aided welding fixture designs, Procedia CIRP (2023). doi:doi.org/10.1016/j.procir.2023.03.143.

[3] I. Osband, J. Aslanides, A. Cassirer, Randomized Prior Functions for Deep Reinforcement Learning, Advances in Neural Information Processing Systems (2018) 8617–8629doi:10.48550/arXiv.1806.03335.

[4] M. Geist, O. Pietquin, Kalman Temporal Differences, Journal of Artificial Intelligence Research 39 (2010) 483–532. doi:doi.org/10.1613/jair.3077.

[5] M. G. Bellemare, W. Dabney, R. Munos, A Distributional Perspective on Reinforcement Learning, 34th International Conference on Machine Learning, ICML 2017 (2017) 693–711.

[6] B. O'Donoghue, I. Osband, R. Munos, V. Mnih, The Uncertainty Bellman Equation and Exploration, 35th International Conference on Machine Learning, ICML 2018 9 (2017) 6154–6173. doi:10.48550/arXiv.1709.05380.

[7] W. R. Thompson, On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples, Biometrika 25 (1933) 285. doi:10.2307/2332286.

[8] B. V. Rosić, A. Litvinenko, O. Pajonk, H. G. Matthies, Sampling-free linear Bayesian update of polynomial chaos representations, Journal of Computational Physics 231 (2012) 5761–5787. doi:10.1016/J.JCP.2012.04.044.

[9] H. G. Matthies, E. Zander, B. V. Rosić, A. Litvinenko, Parameter estimation via conditional expectation: a Bayesian inversion, Advanced Modeling and Simulation in Engineering Sciences 3 (2016) 1–21. doi:10.1186/S40323-016-0075-7.

[10] O. Pajonk, B. V. Rosić, H. G. Matthies, Sampling-free linear Bayesian updating of model state and parameters using a square root approach, Computers & Geosciences 55 (2013) 70–83. doi:10.1016/J.CAGEO.2012.05.017.

[11] M. Geist, O. Pietquin, Algorithmic survey of parametric value function approximation, IEEE Transactions on Neural Networks and Learning Systems 24 (2013) 845–867. doi:10.1109/TNNLS.2013.2247418.

[12] L. Baird, Residual Algorithms: Reinforcement Learning with Function Approximation, Elsevier, 1995. doi:10.1016/B978-1-55860-377-6.50013-X.

[13] Y. Engel, S. Mannor, R. Melr, Bayes Meets Bellman: The Gaussian Process Approach to Temporal Difference Learning, Proceedings of the 20th International Conference on Machine Learning (ICML-03) (2003).

[14] Y. Engel, S. Mannor, R. Meir, Reinforcement learning with Gaussian processes, in: Proceedings of the 22nd International Conference on Machine Learning, ICML '05, Association for Computing Machinery, New York, NY, USA, 2005, p. 201–208. doi:10.1145/1102351.1102377.
URL https://doi.org/10.1145/1102351.1102377

[15] Q. Lu, G. B. Giannakis, Robust and Adaptive Temporal-Difference Learning Using An Ensemble of Gaussian Processes, ArXiv (2021). doi:doi.org/10.48550/arXiv.2112.00882.

[16] J. P. Geerts, K. L. Stachenfeld, N. Burgess, Probabilistic Successor Representations with Kalman Temporal Differences, ArXiv (2019). doi:10.32470/CCN.2019.1323-0.

[17] P. Malekzadeh, M. Salimibeni, A. Mohammadi, A. Assa, K. N. Plataniotis, MM-KTD: Multiple Model Kalman Temporal Differences for Reinforcement Learning, IEEE Access 8 (2020) 128716–128729. doi:10.1109/ACCESS.2020.3007951.

[18] S. D.-C. Shashua, S. Mannor, Deep Robust Kalman Filter, ArXiv (3 2017). doi:10.48550/arxiv.1703.02310.

[19] D. G. Luenberger, Optimization by vector space methods, Wiley, 1968.

[20] B. V. Rosić, A. Litvinenko, O. Pajonk, H. G. Matthies, Sampling-free linear Bayesian update of polynomial chaos representations, Journal of Computational Physics 231 (2012) 5761–5787. doi:10.1016/J.JCP.2012.04.044.

[21] M. L. Puterman, Markov decision processes: Discrete stochastic dynamic programming, Wiley, 2008. doi:10.1002/9780470316887.

[22] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, IEEE Transactions on Neural Networks 9 (1998) 1054–1054. doi:10.1109/TNN.1998.712192.

[23] A. Kolmogorov, Foundations of the Theory of Probability, second english Edition, Dover Publications, 1933.

[24] A. Bobrowski, Functional Analysis for Probability and Stochastic Processes, Cambridge University Press, 2005. doi:10.1017/CBO9780511614583.

[25] A. Antos, C. Szepesvári, R. Munos, Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path, Machine Learning 71 (2008) 89–129. doi:10.1007/S10994-007-5038-2/METRICS.

[26] G. Evensen, Accounting for model errors in iterative ensemble smoothers, Computational Geosciences 23 (2019) 761–775. doi:10.1007/S10596-019-9819-Z/METRICS.

[27] G. Dalal, B. Szörényi, G. Thoppe, S. Mannor, Finite Sample Analyses for TD(0) with Function Approximation, 32nd AAAI Conference on Artificial Intelligence, AAAI 2018 (2017) 6144–6160doi:10.1609/aaai.v32i1.12079.