STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVI**, Number 3, 2011

GEODESIC DISTANCE-BASED KERNEL CONSTRUCTION FOR GAUSSIAN PROCESS VALUE FUNCTION APPROXIMATION

HUNOR JAKAB

Abstract. Finding accurate approximations to state and action value functions is essential in Reinforcement learning tasks on continuous Markov Decision Processes. Using Gaussian processes as function approximators we can simultaneously represent model confidence and generalize to unvisited states. To improve the accuracy of the value function approximation in this article I present a new method of constructing geodesic distance based kernel functions from the Markov Decision process induced graph structure. Using sparse on-line Gaussian process regression the nodes and edges of the graph structure are allocated during on-line learning parallel with the inclusion of new measurements to the basis vector set. This results in a more compact and efficient graph structure and more accurate value function estimates. The approximation accuracy is tested on a simulated robotic control task.

1. Introduction

The majority of real-life control problems including robotic locomotion requires the efficient handling of continuous and high dimensional state and action spaces, therefore function approximation needs to be employed. In real-life control tasks the ability to handle uncertainties arising from noisy measurements is a deciding factor in terms of performance and efficiency. Gaussian processes (GP) can be used efficiently for the approximation of value functions on continuous state spaces. The nonparametric nature of GP's provides increased flexibility and the resulting fully probabilistic model can be used for appropriate uncertainty treatment. One of the major drawbacks of using GP action-value function approximation with Euclidean distance-based kernel functions is the fact that they cannot accurately represent discontinuities. There are many reinforcement learning (RL) tasks where the state or

Received by the editors: April 10, 2011.

2010 *Mathematics Subject Classification.* 68T05, 68T40,60J25.

1998 *CR Categories and Descriptors.* 1.2.6 [**Computing Methodologies**]: Artificial Intelligence – *Learning*; 1.2.9 [**Computing Methodologies**]: Artificial Intelligence – *Robotics*.

Key words and phrases. Reinforcement learning, Gaussian processes.

state-action value function corresponding to the actual policy is discontinuous in some regions of the space. This discontinuity has great influence on the algorithms performance. In this paper I describe a modality to increase the accuracy of our GP action-value function estimates by introducing a new modality of constructing kernel functions that operate on a graph structure induced by the Markov decision process (MDP) underlying the RL problem. Unlike in [7] the nodes of the MDP induced graph structure are allocated dynamically parallel to the inclusion of new basis vectors in the GP estimator. The resulting graph gives a compact representation of the most important points of the state space. Replacing Euclidean distance in the kernel function with distances defined on the paths between data-points from the MDP induced graph leads to more accurate value function approximations.

2. Notation and background

A formal representation of a reinforcement learning problem is given using Markov decision processes [4]. An MDP is a quadruple $M\left(S, A, P, R\right)$ with the following elements: S is the set of states; A the set of actions; $P(s'|s,a) : S \times S \times A \to [0,1]$ the transition probabilities, and $R : S \times A \to \mathbb{R}$, $R(s,a)$ the reward function. Calculating value functions for a policy π is essential for all value-based reinforcement learning algorithms. Value functions measure the long-term usefulness of a given state or state-action pair based on the cumulative reward received when starting out in that state and following a given policy[1] [8]: $V^{\pi}(s) = E_{\pi}\left(\sum_{t=0}^{\infty} \gamma^t r_t | s_t = s\right)$

Different approaches of using Gaussian processes for estimating value functions have been proposed[2, 5, 3] . In our approach we use as training data the states visited during different episodes of the experiment, and the corresponding – possibly – discounted cumulative rewards $\overline{V}(s_t) = \sum_{i=0}^{H-t} \gamma^i r_{t+i}$ as noisy targets.[2] The model is a GP, completely specified by its mean and covariance function, and it can be used to perform regression directly in a function space, with the resulting $\hat{V}(s)$ being the approximation to the state value function. The elements of the kernel matrix $\boldsymbol{K}$ are $\boldsymbol{K}(i,j) = k(s_i, s_j)$, where k is the kernel function operating on state variables. Having processed n points we have a GP built on the data set $\mathcal{D} = [(s_i, \overline{V}_i)]_{i=1,n}$ which is also called the set of basis vectors (BV). To estimate the value of a new state, s_{n+1}, we compute the predictive mean (1) and variance (2) functions conditioned on the data,

[1]For ease of visualization throughout the article I use state-value functions, but the presented methods naturally apply also to state-action value functions.

[2]We assume that the targets have Gaussian noise with equal variance; one can easily use different *known* noise variance.

given by the posterior GP [6]:

$$\hat{V}_{n+1}|\mathcal{D} \sim \mathcal{N}\left(\mu_{n+1}, \mathrm{cov}(s_{n+1})\right)$$

$$\mu_{n+1} = \boldsymbol{k}_{n+1}\boldsymbol{\alpha}_n \tag{1}$$

$$\mathrm{cov}(\hat{V}_{n+1}, q_{n+1}) = k\left(s_{n+1}, s_{n+1}\right) - \boldsymbol{k}_{n+1}\boldsymbol{C}_n\boldsymbol{k}_{n+1}^T, \tag{2}$$

where $\boldsymbol{\alpha}_n$ and $\boldsymbol{C}_n$ are the parameters of the posterior GP:

$$\boldsymbol{\alpha}_n = [\boldsymbol{K}_q^n + \boldsymbol{\Sigma}_n]^{-1}\overline{V}, \quad \boldsymbol{C}_n = [\boldsymbol{K}_q^n + \boldsymbol{\Sigma}_n]^{-1}. \tag{3}$$

with $\boldsymbol{\Sigma}_n = \boldsymbol{\sigma} I_n$ the covariance of the observation noise and $\boldsymbol{k}_{n+1}$ a vector containing the covariances between the new point and the training points:

$$\boldsymbol{k}_{n+1} = [k(s_1, s_{n+1}), \ldots, k(s_n, s_{n+1})]. \tag{4}$$

The above described method leads to good value function estimates when there are no significant discontinuities in the true value function. Figure 1 shows a comparison between TD and GP approximated value functions for the inverted pendulum (sec.4) balancing task in case of a fixed policy.

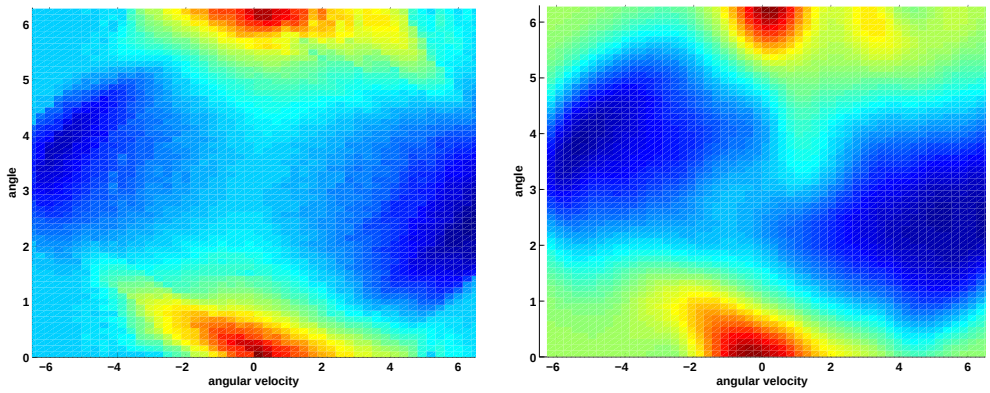

FIGURE 1. Color map of estimated value functions in case of (a) TD approximation , (b) GP approximation

3. GEODESIC DISTANCE BASED ON-LINE KERNEL CONSTRUCTION

To improve the ability of our value function approximator to represent discontinuities a new distance measure for the kernel function k is needed. Let G denote a sparse graph structure induced by the MDP which we will define as follows: G is a graph that has n nodes where $n = |BV|$ is the number of basis vectors present in the GP value function approximator. The connection between these nodes are initialized parallel to the addition of each basis vector to the BV set of the GP. For this procedure sparse on-line GP value function approximation is being used. In this setting only those data-points are added to the BV set which provide significant information gain during the learning

process, thereby reducing the number of the GP parameters drastically. For details of sparse on-line updates consult [1].

Using the GP basis vectors as nodes in our graph construction makes sure that the graph structure remains sparse and the nodes are placed in important regions of the state space. The construction of the MDP induced graph structure during the learning process proceeds as follows: If at time-step t we perform a full update of the GP parameters, adding the data-point s_t to the set of basis vectors, we establish a new node in our graph structure and connect it to the existing graph according to the following rule:

(5)
$$d_{s_t,s_i} = \begin{cases} ED(s_i, s_t) & if s_i = \underset{s_j}{\operatorname{argmin}} \left(\exp\left[-\frac{\|s_{t-j} - s_t\|}{2\sigma_{GP}(s_t)} \right] \right), \\ 0 & \text{otherwise} \end{cases} \quad j = 1 \dots t-1$$

Here d_{s_t,s_i} denotes the connection weight between nodes t and i. I used the notation $ED(\cdot,\cdot)$ to denote the Euclidean distance between two points from the state-action space, and $\sigma_{GP}(s_t)$ to denote the predictive variance of the GP value function approximator at point s_t.

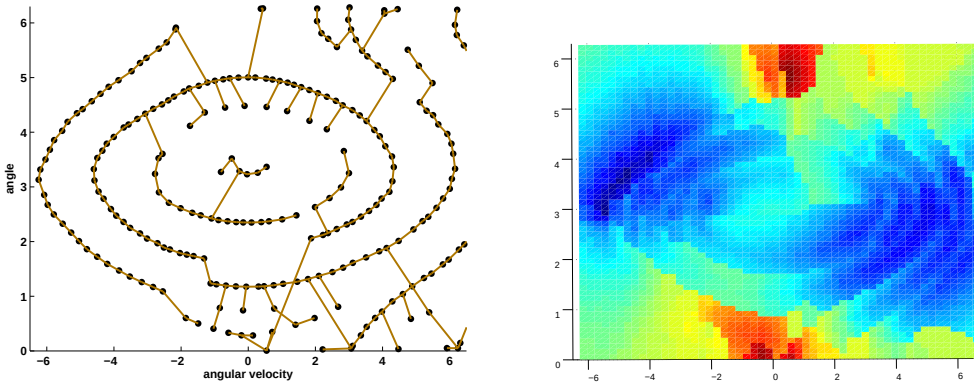

FIGURE 2. (a) Graph structure, (b) GP approx. with geodesic kernel

The small number of graph nodes makes it possible even in case of continuous state-action spaces to represent optimal distances between each node in a lookup table which leads to lesser computational costs. Figure 2.a presents the graph structure obtained after training the GP for a number of episodes on the inverted pendulum problem (sec.4), with a Gaussian policy and a fixed neural-network based controller. Shortest paths between nodes of the graph can be calculated efficiently using Dijkstra's algorithm. Based on this graph structure a new type of kernel function can be built which uses as a distance measure the shortest path between two data-points.

(6)
$$k(s, s') = \exp\left(\frac{SP(s, s')^2}{2\sigma^2} \right)$$

The definition of the shortest path exists only between data-points that are present in the GP basis vector set. In a continuous state-action space visiting the same state-action pair twice has very low probability, therefore we have to define our shortest path measure between two points as the distance between the two basis-vectors that are the closest to the data-points plus the distance of the data-points from these Basis vectors.

$$\begin{aligned} SP(s,s') &= ED(s,s_i) + SP(s_i,s_j) + ED(s_j,s') \\ s_i &= \underset{s_i}{\text{argmax}}(ED(s,s_i)) \quad i = 1 \dots n \\ s_j &= \underset{s_j}{\text{argmax}}(ED(s',s_j)) \quad j = 1 \dots n \end{aligned}$$

The expression for the predictive mean in eq.(1) can be regarded as a weighted linear combination of the value measurements in each basis vector. The weights given by $\boldsymbol{k}_{n+1}$ from eq.(4) using the newly defined covariance function from eq.(6) represent how far each basis vector is located on a sequence of states from the test-point.[3] This can also be regarded as a modified eligibility trace.

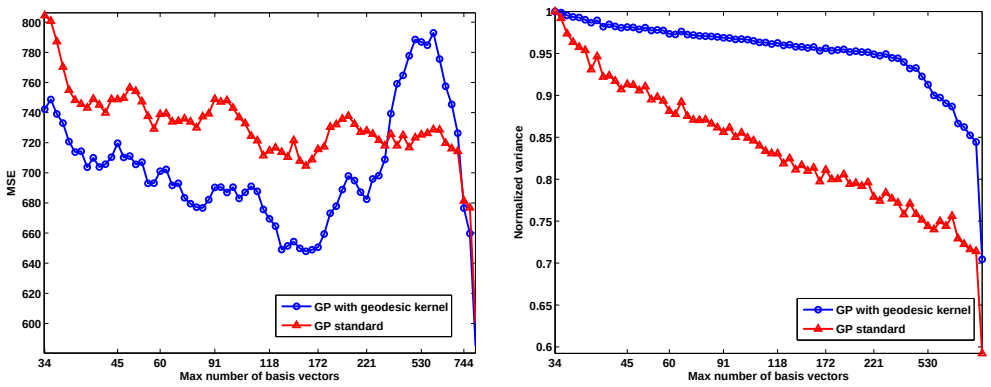

FIGURE 3. (a) Mean squared error, (b) Normalized Variance

4. EXPERIMENTS AND RESULTS

The above presented method was tested on a simulated control problem, the classical pendulum balancing task where both the state and the action spaces are continuous. A state variable $s = (\theta, \omega)$ consists of the angle and angular velocity of the pendulum, actions are the torques that we can apply to

[3]Note that from the definition of the connections between the graph nodes follows that the sequence of states upon which the distance is measured is always executable under the current policy π.

the system, and are limited to a $[-5, 5]$ interval. The performance of the proposed value-function approximation scheme is tested under a fixed Gaussian policy which consists of a deterministic controller and added Gaussian noise with fixed variance $\pi(s, a) = \mathcal{N}(0, \sigma^2) + f_\theta(s)$. As a baseline I used the TD approximation of the corresponding value function, based on 800 episodes of length 150. The state space was discretized to contain 3600 states. Figure 3 shows the approximation accuracy of both standard GP and Geodesic distance based GP value function approximation where the horizontal axis represents the maximum number of allowed basis vectors and the vertical axis measures the mean squared approximation error. In terms of approximation error GP with geodesic Gaussian kernel performs significantly better by low number of basis vectors, and achieves the same performance as standard GP after the number of BV's exceeds a threshold. However the variance of the value function estimates decreases slower when geodesic kernel is being used. There is also a performance decrease in a certain region of max BV numbers where the performance gets worse than standard GP.

5. conclusion

In this article I have presented a modality of dynamically constructing geodesic distance based kernel functions during on-line Gaussian Process value function approximation. Unlike in previous work where fixed resolution graph structures have been used I presented a way to construct the MDP induced graph only between data-points which are important from the information-gain point of view. Experimental results prove the viability of this method.

References

[1] Lehel Csató and Manfred Opper. Sparse on-line Gaussian Processes. *Neural Computation*, 14(3):641–669, 2002.
[2] Marc Peter Deisenroth, Carl Edward Rasmussen, and Jan Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009.
[3] Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with Gaussian processes. In *ICML '05* , pages 201–208, New York, NY, USA, 2005. ACM.
[4] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, 1994.
[5] C. E. Rasmussen and M. Kuss. Gaussian processes in reinforcement learning. In L. K. Saul Thrun, S. and B. Schlkopf, editors, *NIPS 2003*, pages 751–759. MIT Press, 2004.
[6] Carl Edward Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
[7] Masashi Sugiyama, Hirotaka Hachiya, Christopher Towell, and Sethu Vijayakumar. Geodesic gaussian kernels for value function approximation. *Auton. Robots*, 25:287–304.
[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Department of Mathematics and Computer Science, Babeş-Bolyai University, 1 Kogălniceanu St., 400084 Cluj-Napoca, Romania
E-mail address: jakabh@cs.ubbcluj.ro