Feature Selection for Value Function Approximation Using Bayesian Model Selection

Tobias Jung and Peter Stone
Department of Computer Sciences, University of Texas at Austin
{tjung,pstone}@cs.utexas.edu

Motivation

One fundamental problem arising in the policy iteration framework of infinite horizon dynamic programming and reinforcement learning (RL) is approximating the value function under a stationary policy (APE) from a trajectory of sample transitions [2, 14]. Especially for continuous, potentially high-dimensional state spaces, this becomes a rather difficult problem that, at present, has no completely satisfying solution. Despite having powerful and well-understood algorithms for the solution of APE, such as LSTD, LSPE, BRM (that for linearly parameterized approximations come with certain theoretical convergence guarantees and can be solved efficiently in closed-form), deciding which features (basis functions) to use is quite challenging, and in general, needs to be done manually: thus it is tedious, prone to errors, and most important of all, requires considerable insight into the domain. It would be far more desirable if a learning system could automatically choose its own representation. In particular, considering efficiency, we want the system to adapt to the actual difficulties faced, without wasting resources: often, there are many factors that can make a particular problem easier than it initially appears to be, for example, when the input data lies on a low-dimensional submanifold of the input space.

Our approach

Recent work in applying nonparametric function approximation to RL, such as Gaussian processes (GP) [4, 12, 13, 3], or equivalently, regularization networks [6], is a very promising step in this direction. Instead of having to explicitly specify individual basis functions, we have to specify a more general covariance function (or kernel), that only depends on a very small number of hyperparameters and, in a sense, automatically 'generates' basis functions. The key contribution of our research is to demonstrate that then, using any of several possible model selection methods for the hyperparameters, such as marginal likelihood optimization in a Bayesian setting, or leave-one-out (LOO) error minimization in a frequentist setting, the task of feature selection in RL can be automated. In consequence, given just a batch of sample transitions from the process, we can solve the whole APE problem for any learning task fully automatically (without tweaking and adjusting hyperparameters/learning rates, and, in theory, independent of the dimensionality of the state space).

Here we will focus on the Bayesian setting, and adapt marginal likelihood optimization for the GP-based APE method GPTD introduced in [4]. To make the approach feasible for large-scale applications (large-scale meaning large number of data points/state transitions), we employ the SR-approximation [10], which approximates the data-dependent covariance matrix using the product of matrices of vastly reduced size, and reduces overall computational complexity substantially. To obtain this approximation, we use the incomplete Cholesky decomposition (ICD) [5, 1] to identify from the data a small number of relevant basis elements. Furthermore, only by automatic model selection will we be able to use more sophisticated covariance functions, which we will allow us to uncover the "hidden" properties of a given problem. For example, by choosing an RBF kernel with independent lengthscales for the individual dimensions of the state space, model selection will automatically drive those components to zero that correspond to state variables irrelevant (or redundant) to the task. This will allow us to concentrate our computational efforts on the parts of the input space that really matter and will further improve computational efficiency: as we remove redundant variables from the states, the effective rank of the covariance matrix will decrease and, in consequence, fewer basis elements in the SR-approximation will be selected by ICD. As a side-effect, because it is generally easier to learn in "smaller" spaces, this may also benefit generalization and thus help us to reduce sample complexity.

As much of this is ongoing research, we currently do not have extensive results on large-scale real-world applications (where of course APE has to be integrated into the policy iteration framework). But we can illustrate the various benefits of our approach in synthetic small-scale domains.

Related work

Despite its many promises, previous work with GPs in RL rarely explores the benefits of model selection: in [13], a variant of stochastic search was used to determine hyperparameters of the covariance for GPTD

using as score function the online performance of an agent. In [12], standard GPs with marginal likelihood based model selection were employed; however, since their approach was based on fitted value iteration, the task of value function approximation was reduced to ordinary regression.

More closely related to our work is the approach described in [9], which adapts the hyperparameters of RBF-basis functions (both their location and lengthscales) using either gradient descent or the cross-entropy method on the Bellman error. However, because basis functions are adapted individually (and their number is chosen in advance), the method is prone to overfitting: e.g. by placing basis functions with very small width near discontinuities. The problem is compounded when only few data points are available. In contrast, using a Bayesian approach, we can automatically trade-off model fit and model complexity with the number of data points, choosing always the best complexity: e.g. for small data sets we will prefer larger lengthscales (less complex), for larger data sets we can afford smaller lengthscales (more complex).

Other alternative approaches do not rely on predefined basis functions: The method in [7] is an incremental approach that uses dimensionality reduction and state aggregation to create new basis functions such that for every step the remaining Bellman error for a trajectory of states is successively reduced. A related approach is given in [11] which incrementally constructs an orthogonal basis for the Bellman error. A graph-based unsupervised approach is presented in [8], which derives basis functions from the eigenvectors of the graph Laplacian induced from the underlying MDP.

References

[1] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *JMLR*, 3:1–48, 2002.

[2] D. Bertsekas. *Dynamic programming and Optimal Control, Vol. II.* Athena Scientific, 2007.

[3] M. P. Deisenroth, C. E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, x:xx–xx, 2009.

[4] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with Gaussian processes. In *Proc. of ICML 22*, 2005.

[5] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representation. *JMLR*, 2:243–264, 2001.

[6] T. Jung and D. Polani. Learning robocup-keepaway with kernels. *JMLR: Workshop and Conference Proceedings (Gaussian Processes in Practice)*, 1:33–57, 2007.

[7] P. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proc. of ICML 2006*, 2006.

[8] S. Mahadevan and M. Maggioni. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *JMLR*, 8:2169–2231, 2007.

[9] N. Menache, N. Shimkin, and S. Mannor. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 134:215–238, 2005.

[10] J. Qui n onero Candela, C. E. Rasmussen, and C. K. I. Williams. Approximation methods for gaussian process regression. In Leon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large Scale Learning Machines*, pages 203–223. MIT Press, 2007.

[11] R. Parr, C. Painter-Wakefield, L. Li, and M. Littman. Analyzing feature generation for value-function approximation. In *Proc. of ICML 2007*, 2007.

[12] C. E. Rasmussen and M. Kuss. Gaussian processes in reinforcement learning. In *Advances in Neural Information Processing Systems 16*, pages 751–759. MIT Press, 2004.

[13] J. Reisinger, P. Stone, and R. Miikkulainen. Online kernel selection for Bayesian reinforcement learning. In *Proc. of ICML 25*, 2008.

[14] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.