Gaussian Processes for Fast Policy Optimisation of POMDP-based Dialogue Managers

M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu and S. Young
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK
{mg436, fj228, sk561, farm2, brmt2, ky219, sjy}@eng.cam.ac.uk

Abstract

Modelling dialogue as a Partially Observable Markov Decision Process (POMDP) enables a dialogue policy robust to speech understanding errors to be learnt. However, a major challenge in POMDP policy learning is to maintain tractability, so the use of approximation is inevitable. We propose applying Gaussian Processes in Reinforcement learning of optimal POMDP dialogue policies, in order (1) to make the learning process faster and (2) to obtain an estimate of the uncertainty of the approximation. We first demonstrate the idea on a simple voice mail dialogue task and then apply this method to a real-world tourist information dialogue task.

1 Introduction

One of the main challenges in dialogue management is effective handling of speech understanding errors. Instead of hand-crafting the error handler for each dialogue step, statistical approaches allow the optimal dialogue manager behaviour to be learnt automatically. Reinforcement learning (RL), in particular, enables the notion of planning to be embedded in the dialogue management criteria. The objective of the dialogue manager is for each dialogue state to choose such an action that leads to the highest expected long-term reward, which is defined in this framework by the Q-function. This is in contrast to Supervised learning, which estimates a dialogue strategy in such a way as to make it resemble the behaviour from a given corpus, but without directly optimising overall dialogue success.

Modelling dialogue as a Partially Observable Markov Decision Process (POMDP) allows action selection to be based on the differing levels of uncertainty in each dialogue state as well as the overall reward. This approach requires that a distribution of states (*belief state*) is maintained at each turn. This explicit representation of uncertainty in the POMDP gives it the potential to produce more robust dialogue policies (Young et al., 2010).

The main challenge in the POMDP approach is the tractability of the learning process. A discrete state space POMDP can be perceived as a continuous space MDP where the state space consists of the belief states of the original POMDP. A grid-based approach to policy optimisation assumes discretisation of this space, allowing for discrete space MDP algorithms to be used for learning (Brafman, 1997) and thus approximating the optimal Q-function. Such an approach takes the order of $100,000$ dialogues to train a real-world dialogue manager. Therefore, the training normally takes place in interaction with a simulated user, rather than real users. This raises questions regarding the quality of the approximation as well as the potential discrepancy between simulated and real user behaviour.

Gaussian Processes have been successfully used in Reinforcement learning for continuous space MDPs, for both model-free approaches (Engel et al., 2005) and model-based approaches (Deisenroth et al., 2009). We propose using GP Reinforcement learning in a POMDP dialogue manager to, firstly, speed up the learning process and, secondly, obtain the uncertainty of the approximation. We opt for the model-free approach since it has the potential to allow the policy obtained in interaction with the simulated user to be further refined in interaction with real users.

In the next section, the core idea of the method is explained on a toy dialogue problem where different aspects of GP learning are examined. Following that, in Section 3, it is demonstrated how this methodology can be effectively applied to a real world dialogue. We conclude with Section 4.

2 Gaussian Process RL on a Toy Problem

2.1 Gaussian Process RL

A Gaussian Process is a generative model of Bayesian inference that can be used for function regression (Rasmussen and Williams, 2005). A Gaussian Process is fully defined by a mean and a kernel function. The kernel function defines prior function correlations, which is crucial for obtaining good posterior estimates with just a few observations. GP-Sarsa is an on-line reinforcement learning algorithm for both continuous and discrete MDPs that incorporates GP regression (En-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 201–204,
The University of Tokyo, September 24-25, 2010.

gel et al., 2005). Given the observation of rewards, it estimates the Q-function utilising its correlations in different parts of the state and the action space defined by the kernel function. It also gives a variance of the estimate, thus modelling the uncertainty of the approximation.

2.2 Voice Mail Dialogue Task

In order to demonstrate how this methodology can be applied to a dialogue system, we first explain the idea on the voice mail dialogue problem (Williams, 2006).

The state space of this task consists of three states: the user asked for the message either to be saved or deleted, or the dialogue ended. The system can take three actions: ask the user what to do, save or delete the message. The observation of what the user wants is corrupted with noise, therefore we model this as a three-state POMDP. This POMDP can be viewed as a continuous MDP, where the MDP state is the POMDP belief state, a 3-dimensional vector of probabilities. For both learning and evaluation, a simulated user is used which makes an error with probability 0.3 and terminates the dialogue after at most 10 turns. In the final state, it gives a positive reward of 10 or a penalty of -100 depending on whether the system performed a correct action or not. Each intermediate state receives the penalty of -1. In order to keep the problem simple, a model defining transition and observation probabilities is assumed so that the belief can be easily updated, but the policy optimisation is performed in an on-line fashion.

2.3 Kernel Choice for GP-Sarsa

The choice of kernel function is very important since it defines the prior knowledge about the Q-function correlations. They have to be defined on both states and actions. In the voice mail dialogue problem the action space is discrete, so we opt for a simple δ kernel over actions:

$$k(a, a') = 1 - \delta_a(a'), \qquad (1)$$

where δ_a is the Kronecker delta function. The state space is a 3-dimensional continuous space and the kernel functions over the state space that we explore are given in Table 1. Each kernel function defines a different correlation. The polynomial kernel views elements of the state vector as features, the dot-product of which defines the correlation. They can be given different relevance r_i in the parametrised version. The Gaussian kernel accounts for smoothness, *i.e.*, if two states are close to each other the Q-function in these states is correlated. The scaled norm kernel defines positive correlations in the points that are close to each other and a negative correlation otherwise. This is particularly useful for the voice mail problem, where, if two belief states are very different, taking the same action in these states generates a negatively correlated reward.

kernel function	expression
polynomial	$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$
parametrised poly.	$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{D} \frac{x_i x_i'}{r_i^2}$
Gaussian	$k(\mathbf{x}, \mathbf{x}') = p^2 \exp \frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma_k^2}$
scaled norm	$k(\mathbf{x}, \mathbf{x}') = 1 - \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\|\mathbf{x}\|^2 \|\mathbf{x}'\|^2}$

Table 1: Kernel functions

2.4 Optimisation of Kernel Parameters

Some kernel functions are in a parametrised form, such as Gaussian or parametrised polynomial kernel. These parameters, also called *the hyper-parameters*, are estimated by maximising the marginal likelihood[1] on a given corpus (Rasmussen and Williams, 2005). We adapted the available code (Rasmussen and Williams, 2005) for the Reinforcement learning framework to obtain the optimal hyper-parameters using a dialogue corpus labelled with states, actions and rewards.

2.5 Grid-based RL Algorithms

To assess the performance of GP-Sarsa, it was compared with a standard grid-based algorithm used in (Young et al., 2010). The grid-based approach discretises the continuous space into regions with their representative points. This then allows discrete MDP algorithms to be used for policy optimisation, in this case the Monte Carlo Control (MCC) algorithm (Sutton and Barto, 1998).

2.6 Optimal POMDP Policy

The optimal POMDP policy was obtained using the POMDP solver toolkit (Cassandra, 2005), which implements the Point Based Value Iteration algorithm to solve the POMDP off-line using the underlying transition and observation probabilities. We used 300 sample dialogues between the dialogue manager governed by this policy and the simulated user as data for optimisation of the kernel hyper-parameters (see Section 2.4).

2.7 Training set-up and Evaluation

The dialogue manager was trained in interaction with the simulated user and the performance was compared between the grid-based MCC algorithm and GP-Sarsa across different kernel functions from Table 1.

The intention was, not only to test which algorithm yields the best policy performance, but also to examine the speed of convergence to the optimal policy. All the algorithms use an ϵ-greedy approach where the exploration rate ϵ was fixed at 0.1. The learning process greatly depends on

[1] Also called *evidence maximisation* in the literature.

the actions that are taken during exploration. If early on during the training, the systems discovers a path that generates high rewards due to a lucky choice of actions, then the convergence is faster. To alleviate this, we adopted the following procedure. For every training set-up, exactly the same training iterations were performed using 1000 different random generator seedings. After every 20 dialogues the resulting 1000 partially optimised policies were evaluated. Each of them was tested on 1000 dialogues. The average reward of these 1000 dialogues provides just one point in Fig. 1.

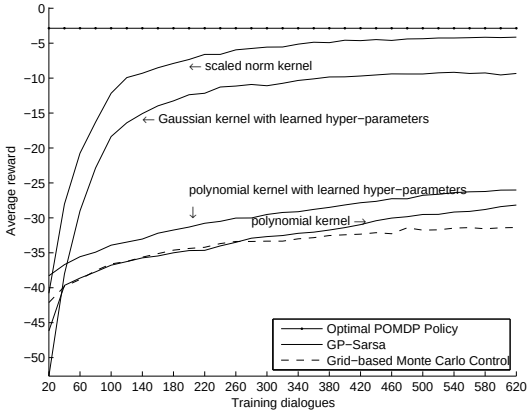

Figure 1: Evaluation results on Voice Mail task

The grid-based MCC algorithm used a Euclidian distance to generate the grid by adding every point that was further than 0.01 from other points as a representative of a new region. As can be seen from Fig 1, the grid-Based MCC algorithm has a relatively slow convergence rate. GP-Sarsa with the polynomial kernel exhibited a learning rate similar to MCC in the first 300 training dialogues, continuing with a more upward learning trend. The parametrised polynomial kernel performs slightly better. The Gaussian kernel, however, achieves a much faster learning rate. The scaled norm kernel achieved close to optimal performance in 400 dialogues, with a much higher convergence rate then the other methods.

3 Gaussian Process RL on a Real-world Task

3.1 HIS Dialogue Manager on CamInfo Domain

We investigate the use of GP-Sarsa in a real-world task by extending the Hidden Information State (HIS) dialogue manager (Young et al., 2010). The application domain is tourist information for Cambridge, whereby the user can ask for information about a restaurant, hotel, museum or another tourist attraction in the local area. The database consists of more than 400 entities each of which has up to 10 attributes that the user can query.
The HIS dialogue manager is a POMDP-based dialogue manager that can tractably maintain belief states for large domains. The key feature of this approach is the grouping of possible user goals into *partitions*, using relationships between different attributes from possible user goals. Partitions are combined with possible user dialogue actions from the N-best user input as well as with the dialogue history. This combination forms the state space – the set of *hypotheses*, the probability distribution over which is maintained during the dialogue. Since the number of states for any real-world problem is too large, for tractable policy learning, both the state and the action space are mapped into smaller scale summary spaces. Once an adequate summary action is found in the summary space, it is mapped back to form an action in the original *master space*.

3.2 Kernel Choice for GP-Sarsa

The summary state in the HIS system is a four-dimensional space consisting of two elements that are continuous (the probability of the top two hypotheses) and two discrete elements (one relating the portion of the database entries that matches the top partition and the other relating to the last user action type). The summary action space is discrete and consists of eleven elements.
In order to apply the GP-Sarsa algorithm, a kernel function needs to be specified for both the summary state space and the summary action space. The nature of this space is quite different from the one described in the toy problem. Therefore, applying a kernel that has negative correlations, such as the scaled norm kernel (Table 1) might give unexpected results. More specifically, for a given summary action, the mapping procedure finds the most appropriate action to perform if such an action exists. This can lead to a lower reward if the summary action is not adequate but would rarely lead to negatively correlated rewards. Also, parametrised kernels could not be used for this task, since there was no corpus available for hyper-parameter optimisation. The polynomial kernel (Table 1) assumes that the elements of the space are features. Due to the way the probability is maintained over this very large state space, the continuous variables potentially encode more information than in the simple toy problem. Therefore, we used the polynomial kernel for the continuous elements. For discrete elements, we utilise the δ-kernel (Eq. 2.3).

3.3 Active Learning GP-Sarsa

The GP RL framework enables modelling the uncertainty of the approximation. The uncertainty estimate can be used to decide which actions to take during the exploration (Deisenroth et al.,

2009). In detail, instead of a random action, the action in which the Q-function for the current state has the highest variance is taken.

3.4 Training Set-up and Evaluation

Policy optimisation is performed by interacting with a simulated user on the dialogue act level. The simulated user gives a reward at the final state of the dialogue, and that is 20 if the dialogue was successful, 0 otherwise, less the number of turns taken to fulfil the user goal. The simulated user takes a maximum of 100 turns in each dialogue, terminating it when all the necessary information has been obtained or if it looses patience.
A grid-based MCC algorithm provides the baseline method. The distance metric used ensures that the number of regions in the grid is small enough for the learning to be tractable (Young et al., 2010).
In order to measure how fast each algorithm learns, a similar training set-up to the one presented in Section 2.7 was adopted and the averaged results are plotted on the graph, Fig. 2.

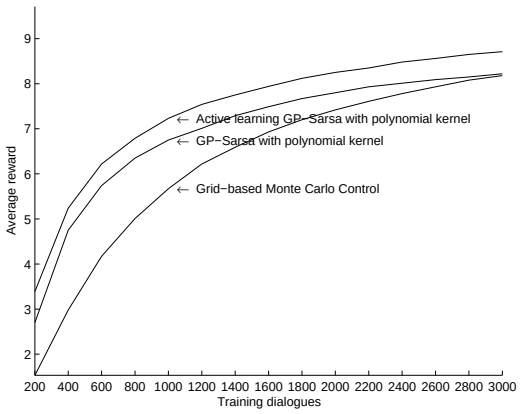

Figure 2: Evaluation results on CamInfo task

The results show that in the very early stage of learning, *i.e.*, during the first 400 dialogues, the GP-based method learns faster. Also, the learning process can be accelerated by adopting the active learning framework where the actions are selected based on the estimated uncertainty.
After performing many iterations in an incremental noise learning set-up (Young et al., 2010) both the GP-Sarsa and the grid-based MCC algorithms converge to the same performance.

4 Conclusions

This paper has described how Gaussian Processes in Reinforcement learning can be successfully applied to dialogue management. We implemented a GP-Sarsa algorithm on a toy dialogue problem, showing that with an appropriate kernel function faster convergence can be achieved. We also demonstrated how kernel parameters can be learnt from a dialogue corpus, thus creating a bridge between Supervised and Reinforcement learning methods in dialogue management. We applied GP-Sarsa to a real-world dialogue task showing that, on average, this method can learn faster than a grid-based algorithm. We also showed that the variance that GP is estimating can be used in an Active learning setting to further accelerate policy optimisation.
Further research is needed in the area of kernel function selection. The results here suggest that the GP framework can facilitate faster learning, which potentially allows the use of larger summary spaces. In addition, being able to learn efficiently from a small number of dialogues offers the potential for learning from direct interaction with real users.

Acknowledgements

The authors would like to thank Carl Rasmussen for valuable discussions. This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and by the EU FP7 Programme under grant agreement 216594 (CLASSiC project).

References

RI Brafman. 1997. A Heuristic Variable Grid Solution Method for POMDPs. In *AAAI*, Cambridge, MA.

AR Cassandra. 2005. POMDP solver. `http://www.cassandra.org/pomdp/code/index.shtml`.

MP Deisenroth, CE Rasmussen, and J Peters. 2009. Gaussian Process Dynamic Programming. *Neurocomput.*, 72(7-9):1508–1524.

Y Engel, S Mannor, and R Meir. 2005. Reinforcement learning with Gaussian processes. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 201–208, New York, NY.

CE Rasmussen and CKI Williams. 2005. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

RS Sutton and AG Barto. 1998. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.

JD Williams. 2006. *Partially Observable Markov Decision Processes for Spoken Dialogue Management*. Ph.D. thesis, University of Cambridge.

SJ Young, M Gašić, S Keizer, F Mairesse, J Schatzmann, B Thomson, and K Yu. 2010. The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.