On-line policy optimisation of spoken dialogue systems via live interaction with human subjects

Milica Gašić, Filip Jurčíček, Blaise Thomson, Kai Yu and Steve Young

Cambridge University Engineering Department
Trumpington St, CB1 2PZ, Cambridge, UK
{mg436,fj228,brmt2,ky219,sjy}@eng.cam.ac.uk

***Abstract*—Statistical dialogue models have required a large number of dialogues to optimise the dialogue policy, relying on the use of a simulated user. This results in a mismatch between training and live conditions, and significant development costs for the simulator thereby mitigating many of the claimed benefits of such models. Recent work on Gaussian process reinforcement learning, has shown that learning can be substantially accelerated. This paper reports on an experiment to learn a policy for a real-world task directly from human interaction using rewards provided by users. It shows that a usable policy can be learnt in just a few hundred dialogues without needing a user simulator and, using a learning strategy that reduces the risk of taking bad actions. The paper also investigates adaptation behaviour when the system continues learning for several thousand dialogues and highlights the need for robustness to noisy rewards.**

I. Introduction

The statistical approach to dialogue modelling has been proposed as a means of building domain independent dialogue systems, trainable from data and robust to speech understanding errors [1], [2]. If the dialogue state satisfies the Markov property, the dialogue can be modelled as a Markov decision process (MDP) [1] and reinforcement learning (RL) algorithms can be used for policy optimisation [3]. Since RL is typically slow, policy training in the past has normally required the use of a simulated user [4], and where direct human-computer interaction has been attempted, as in the NJFun system [5], the dialogue systems have been constrained and reliant on a significant amount of built-in expert knowledge.

A recent trend has been to move to the partially observable Markov decision process (POMDP) in order to provide increased robustness to errors in speech understanding [6], [7]. The POMDP-based approach to dialogue management maintains a distribution over every possible dialogue state, the *belief state* and based on that distribution, the system chooses the action that gives the highest expected reward. Various approximations allow this method to be used for building real world dialogue systems [8], [9]. However, POMDP systems are more complex than MDP systems and they typically require $\mathcal{O}(10^5)$ dialogues [10] to train using conventional RL algorithms. This makes it prohibitive to train in direct interaction with human users and the use of a simulated user appears essential despite the disadvantages of additional development costs and potential discrepancies between real and simulated user behaviour.

Gaussian process (GP) based RL [11] has been recently applied to POMDP dialogue policy optimisation in order to exploit the correlations between different belief states and thus speed up the learning process [12]. GP also provides an estimate of the uncertainty of the approximation which can be used to obtain more efficient learning strategies [13]. Furthermore, recent innovations in crowd-sourcing and global telephone call routing via VoIP now allow large numbers of users to be recruited at low cost for large-scale training and testing of dialogue systems [14].

This paper reports on an experiment to learn a dialogue policy for a real-world task directly from human interaction using a binary reward signal provided by users at the end of each dialogue. The domain is the Cambridge tourist information for restaurants, pubs and bars, which contains about 400 venues each of which has up to ten attributes that the user may query, and the dialogue system is the POMDP-based Hidden Information State system. Using GP-Sarsa, it is shown that a usable policy can be learnt from scratch in just a few hundred dialogues without needing a user simulator for bootstrapping and, using a learning strategy that reduces the risk of taking bad actions. In the second part of the paper, the behaviour of the system is investigated when on-line learning is allowed to continue for several thousand dialogues. In this case, some interesting phenomena are observed. In particular, the need for robustness to errors in the reward signal is highlighted.

The rest of the paper is organised as follows. Section II briefly reviews the Hidden Information State system and the Gaussian process approach to reinforcement learning. Section III then presents a learning strategy which reduces the risk of taking bad actions during training and is therefore particularly well-suited for learning on-line with real users. Section IV describes the experimental set-up and the results of the initial on-line learning using a few hundred dialogues and the longer-term adaptation using a few thousand dialogues. Finally, conclusions are given in Section V.

II. Background

A. Hidden Information State system

The Hidden Information State (HIS) [8] system is a scalable POMDP-based dialogue system able to sustain real time collaborative dialogues with real users [8], [15]. It achieves its operational efficiency by merging similar dialogue states together. To achieve tractable policy learning, both the belief state and the action space are mapped into smaller scale

978-1-4673-0367-5/11/$26.00 ©2011 IEEE

summary spaces. The summary state is a four dimensional space consisting of two elements that are continuous (the probability of the top two dialogue states) and two discrete elements (one relating the proportion of database entries that match the top dialogue state and the other relating to the last user action type). The summary action space consists of eleven basic actions.

B. Gaussian processes in POMDP dialogue policy optimisation

The role of a dialogue policy π is to map each summary state $\mathbf{b}$ into a summary action a so as to maximise the expected cumulative reward defined by the Q-function as:

$$Q(\mathbf{b},a) = \max_{\pi} E_{\pi}\left(\sum_{\tau=t+1}^{T} \gamma^{\tau-t-1} r_{\tau} | \mathbf{b}_t = \mathbf{b}, a_t = a\right), \quad (1)$$

where r_τ is the reward obtained at time τ, T is the dialogue length and γ is the discount factor, $0 < \gamma \leq 1$.

A Gaussian process (GP) is a generative model of Bayesian inference that can be used for function regression [16]. It is fully defined by a mean and a kernel function which defines prior function correlations and is crucial for obtaining good posterior estimates with just a few observations. GP-Sarsa is an on-line RL algorithm that models the Q-function as a zero mean Gaussian process [17] which defines correlations in different parts of the summary state and action spaces through a kernel function, $Q(\mathbf{b},a) \sim \mathcal{GP}\left(0, k((\mathbf{b},a),(\mathbf{b},a))\right)$ where the kernel $k(\cdot,\cdot)$ is factored into separate kernels over the summary state and action spaces $k_{\mathcal{B}}(\mathbf{b},\mathbf{b})k_{\mathcal{A}}(a,a)$.

For a sequence of summary state-action pairs $\mathbf{B}_t = [(\mathbf{b}^0,a^0),\ldots,(\mathbf{b}^t,a^t)]^{\mathsf{T}}$ visited in a dialogue and the corresponding observed immediate rewards $\mathbf{r}_t = [r^1,\ldots,r^t]^{\mathsf{T}}$, the posterior of the Q-function for any summary state-action pair $(\mathbf{b},a)$ is defined by the following:

$$\begin{aligned}
&Q(\mathbf{b},a)|\mathbf{r}_t,\mathbf{B}_t \sim \mathcal{N}(\overline{Q}(\mathbf{b},a), cov((\mathbf{b},a),(\mathbf{b},a))),\\
&\overline{Q}(\mathbf{b},a) = \mathbf{k}_t(\mathbf{b},a)^{\mathsf{T}}\mathbf{H}_t^{\mathsf{T}}(\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^{\mathsf{T}} + \sigma^2\mathbf{H}_t\mathbf{H}_t^{\mathsf{T}})^{-1}\mathbf{r}_t,\\
&cov((\mathbf{b},a),(\mathbf{b},a)) = k((\mathbf{b},a),(\mathbf{b},a))\\
&\quad -\mathbf{k}_t(\mathbf{b},a)^{\mathsf{T}}\mathbf{H}_t^{\mathsf{T}}(\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^{\mathsf{T}} + \sigma^2\mathbf{H}_t\mathbf{H}_t^{\mathsf{T}})^{-1}\mathbf{H}_t\mathbf{k}_t(\mathbf{b},a)\\
&\mathbf{H}_t = \begin{bmatrix} 1 & -\gamma & \cdots & 0 & 0\\ 0 & 1 & \cdots & 0 & 0\\ \vdots & \ddots & \ddots & \vdots & \vdots\\ 0 & \cdots & 0 & 1 & -\gamma \end{bmatrix},\\
&\mathbf{K}_t(\mathbf{b},a) = [\mathbf{k}_t((\mathbf{b}^0,a^0)),\ldots,\mathbf{k}_t((\mathbf{b}^t,a^t))],\\
&\mathbf{k}_t(\mathbf{b},a) = [k((\mathbf{b}^0,a^0),(\mathbf{b},a)),\ldots,k((\mathbf{b}^t,a^t),(\mathbf{b},a))]^{\mathsf{T}}
\end{aligned} \quad (2)$$

where σ^2 is the additive noise variance in the estimate of the reward such that the marginal likelihood of the observed rewards is modelled by

$$\mathbf{r}_t|\mathbf{B}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{H}_t(\mathbf{K}_t + \sigma^2\mathbf{I})\mathbf{H}_t^{\mathsf{T}}). \quad (3)$$

III. GP STOCHASTIC POLICY FOR LOW RISK LEARNING

On-line reinforcement learning algorithms are often based on ϵ-greedy learning [3] whereby at each turn a random action is taken with probability ϵ otherwise the action with the highest expected Q-value is taken; these actions respectively constitute *exploration* and *exploitation*. Such a learning strategy, however, is not well-suited for learning with real users, especially customers, since it allows any action to be taken during exploration, even ones which are known to lead to poor performance. This can be mitigated to some extent by using hand-crafted rules to define the set of permissible actions for every summary state [18]. Alternatively, using GP-Sarsa the estimate of the variance for each summary state-action pair can be used to focus on actions which appear to be useful but whose benefit is currently uncertain. This has the added benefit that learning rates are also improved [12], [13]. There is the drawback, however, that these methods require manual setting of one or more tuning parameters.

In this paper we propose an alternative stochastic approach that automatically balances exploration and exploitation without the need for hand-crafting or additional parameters. Since the Gaussian process for the Q-function defines a Gaussian distribution for every summary state-action pair (Eq. 2), when a new summary point $\mathbf{b}$ is encountered, for each action $a_i \in \mathcal{A}$, there is a Gaussian distribution $Q(\mathbf{b},a_i) \sim \mathcal{N}(\overline{Q}(\mathbf{b},a_i), cov((\mathbf{b},a_i),(\mathbf{b},a_i)))$. Sampling from these Gaussian distributions gives a set of Q-values for each action $\{Q^i(\mathbf{b},a_i)\}$ from which the action with the highest sampled Q-value can be selected:

$$a = \arg\max_{a_i} Q^i(\mathbf{b},a_i). \quad (4)$$

Thus, this method maps the GP approximation of the Q-function into a stochastic policy which does not require manual balancing of exploration and exploitation.

The effectiveness of this learning strategy can be evaluated by comparing it with an ϵ-greedy policy using the variance exploration method as in [12]. A second order polynomial kernel is used over belief space [19], since this has been shown previously to give good performance on this task [20], and the action space kernel is a simple δ-function. 1000 training sessions with different random seeds were conducted with a simulated user. A reward of 20 was given in the final state of the dialogue if the dialogue was successful, 0 otherwise, less the number of turns taken to fulfil the user goal.

After every batch of 200 training dialogues, the partially trained policies were evaluated on 1000 simulated dialogues. In the case of the ϵ-greedy policy the exploration was switched off during the evaluation. Results are given in Fig. 1, where it can be seen that the stochastic policy learns with a reduced variability in the reward, thus reducing the risk of taking bad actions during learning.

The primary performance metric for a spoken dialogue system applied to an information seeking application is the average success rate, where success is defined as conveying to the user the information that they require. For a single deployed dialogue system which is learning sequentially from a succession of users, it is not possible to compute an average success rate after each dialogue, so here a moving average is used. Fig. 2 shows the moving average success rate of

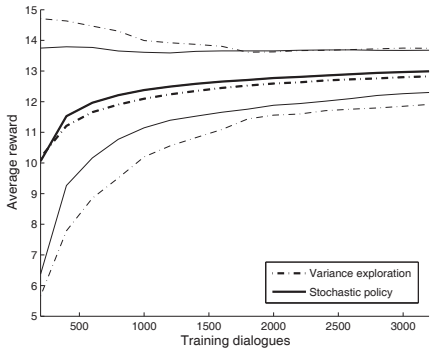

Fig. 1. Average reward vs training epoch with a simulated user for a stochastic policy compared to an ϵ-greedy policy with variance exploration. The upper and lower bounds in each case denote 95% confidence intervals.

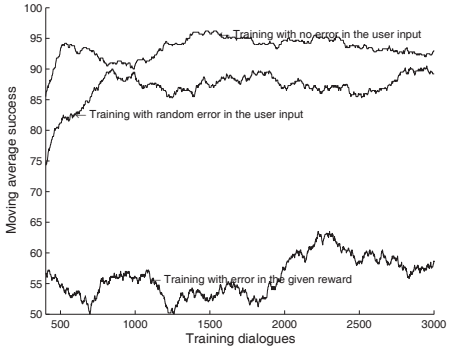

Fig. 3. Moving average success rate for different training conditions: training with no input or reward errors (upper curve); training with 0% to 50% input errors (middle curve); training with 30% reward errors (lower curve).

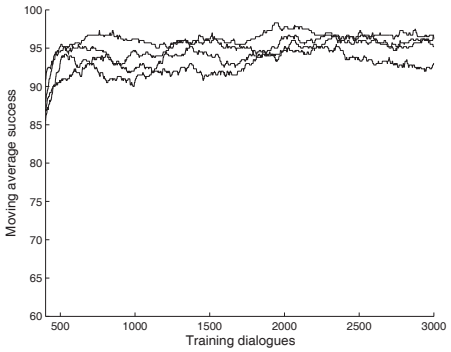

Fig. 2. Moving average success rate of four training sessions of the stochastic policy using the user simulator. The average is computed over the preceding 400 dialogues.

the stochastic policy during training with the user simulator using a window formed from the previous 400 dialogues. This is repeated four times using different random seeds of the simulator to show the variability. There is no statistical difference in the performance, yet as can be seen, there are considerable fluctuations in the trajectories.

Figs 1 and 2 were computed assuming that there are no errors in understanding the user input, or computing the reward function. Fig. 3 compares results when an error is inserted into the user input with random error rate between 0 and 50% and when an inaccurate reward signal is given to the dialogue manager 30% of the time. As can be seen, the reduction in performance caused by errors in the user input is relatively small compared to the reduction caused by inaccurate rewards. The system is therefore relatively robust to understanding errors but is very sensitive to errors in the reward function.

IV. On-line learning with humans

A. Experimental set-up

The stochastic learning strategy presented in the previous section was implemented in a live telephone-based spoken dialogue system in which human users were assigned specific tasks in the Cambridge Restaurant domain using the Amazon Mechanical Turk service [21] in a similar set-up as in [14]. Dialogue tasks were randomly generated and each user could perform up to 40 tasks. At the end of each call, users were asked to press 1 if they were satisfied (i.e. believed that they had been successful in fulfilling the assigned task) and 0 otherwise. This rating was used to compute the reward function described above. The system was initialised with the random policy and then optimised after each dialogue.

The experiment lasted 24 days during which 2960 dialogues with user feedbacks were collected and 252 users took part. To give a flavour of the dialogues and the learning process, a typical task is shown in Table I, Table II then shows an unsuccessful dialogue conducted early in the trial, and Table III shows a successful example 7 days later.

TABLE I
EXAMPLE DIALOGUE TASK

You want to find a pub and it should allow children and have TV. Make sure you get the address, phone number, and postcode.

B. Initial policy optimisation

The performance achieved during on-line learning on the initial 680 dialogues was compared to the performance of a random policy and a policy trained to convergence on the user simulator [10]. For the on-line learning case, the moving average success rate was calculated over a moving window of 400 dialogues, using the user feedback as the indicator of dialogue success. For the random policy and the policy trained on the user simulator, the average success rate was calculated from the user feedback during separate trials of 114 and 466 dialogues, respectively. The results are given in Fig. 4, where the shaded area represents the 95% confidence interval. It can be seen that the on-line learning not only significantly outperforms the random policy, but, in only 680 dialogues achieves a performance which is statistically insignificantly

TABLE II
UNSUCCESSFUL DIALOGUE RECORDED ON 29TH MAY

S	How may I help you?
U	Silence
S	How may I help you?
U	I'm looking for a pub that has TV.
S	The Avery has a television and it's a great pub.
U	Does it allow children?
S	The phone number of The Avery is 01223323405.
U	Does it allow children?
S	Are you looking for something with children?
U	Yes.
S	Can I help you with anything else?
U	Silence
S	Can I help you with anything else?
U	No, thank you good bye.
S	Let me confirm. You are looking for a French restaurant.
U	Thank you, good bye.

TABLE III
SUCCESSFUL DIALOGUE RECORDED ON 4TH JUNE

S	How may I help you?
U	I want to find a pub.
S	The Baron of Beef is an outstanding pub.
U	Does it allow children and has TV?
S	The Avery, which allows children, is a great pub with a television.
U	May I have the address, phone number and post-code?
S	The Avery, which is in the area with the postcode CB2 1AB, is on 69-73 Regent Street. Its phone number is 01223323405.
U	Thank you, good bye.

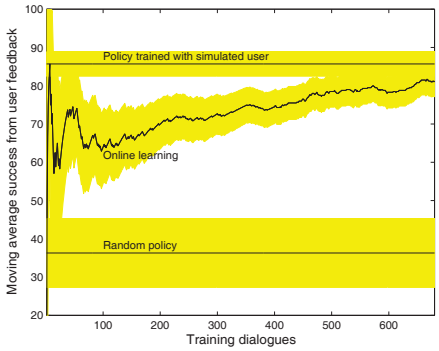

Fig. 4. Initial training stage

Fig. 5. Subjective and objective performance of during adaptation over 2960 dialogues.

different to that of the policy trained with the simulated user, but of course, without the substantial development cost of building a user simulator. Furthermore, when the word error rates of the speech recogniser are compared, see Table IV, it can be seen that speech recognition was a little worse during the on-line learning suggesting that both of the trained policies are comparable in performance.

TABLE IV
COMPARISON BETWEEN WORD ERROR RATES FOR DIFFERENT CORPORA

	Word error rate
Simulator trained policy	20.85
Online learning	22.93
Random policy	25.22

C. Longer term adaptation

In a second phase to the on-line learning experiment, the trial was continued until 2960 dialogues had been processed, with the policy continuing to be adapted after every dialogue. Surprisingly, a long-term cyclic fluctuation was observed with a period of around 1500 dialogues. This is shown in Fig. 5 which also shows two objective measures of success. The first is the partial completion rate in which a task is deemed successful if the system provides the name of the venue that matches the assigned task (e.g. a pub that has a TV and allows children), but does not necessarily provide all the required additional information (e.g. phone, post code, address, etc). The second objective measure is full completion in which a task is deemed to be successful only if the system provides all of the information specified in the assigned task.

It is important to note that neither subjective nor objective task completion rates are 100% accurate since they both depend on the user following the task instructions and asking for all required information. The subjective measure is further confounded by users forgetting or confusing what the task required and therefore incorrectly assessing whether or not the dialogue was successful. As expected, the full completion rate is lower than the partial completion rate and these measures are strongly correlated (see Fig. 5). However, subjective success based on the user feedback is correlated with the objective measures only in the initial stage of learning. Around dialogue 1500 the subjective and the objective measures diverge.

There are two aspects that need to be considered when analysing these results: first is the word error rate, second is the accuracy of the user rating.

Fig. 6 gives the moving average word error rate computed over a window of 400 dialogues. It can be seen that the word error rate in the later stages of learning is higher than in the initial learning stage. To see if this increase in the

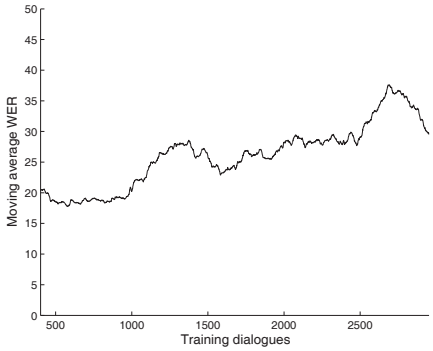

Fig. 6. Moving average word error rate

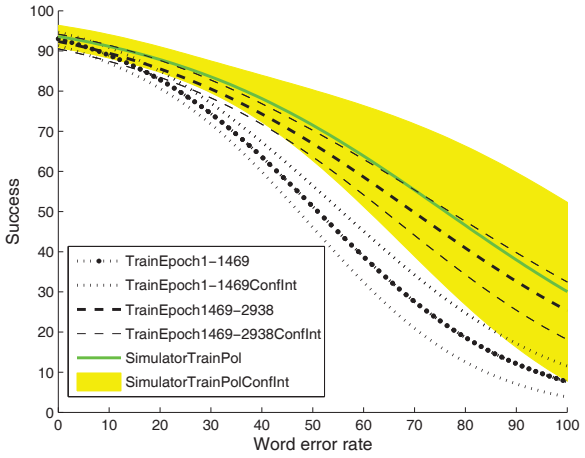

Fig. 7. Subjective success rates during dialogues 1-1469 and dialogues1469-2938 compared with a policy trained on the simulated user. The regions marked as confint denote 95% confidence intervals

word error rate might account for the drop in performance, the corpus was split into two sequential batches and logistic regression used to predict the subjective success rate as a function of the word error rate. The performance of the policies learned on-line in the two batches (TrainEpoch1-1469 and TrainEpoch1469-2938, respectively) are compared to the policy trained on the simulated user; the results are given in Fig. 7. As can be seen, performance on the second training batch is significantly more robust than the first batch and it is statistically indistinguishable from the performance of the policy trained on the simulated user.

However, these results still do not explain the inconsistency between the subjective and the objective measures. More insight into this problem can be gained from the following empirical probabilities:

- $p(\text{feedbck}= 1|\text{complt}= 1)$ – the probability of the user rating the dialogue as successful given that the dialogue task was fully completed, and
- $p(\text{feedbck}= 1|\text{complt}= 0)$ – the probability of the user rating the dialogue as successful even though the dialogue task was not fully completed.

These empirical probabilities were computed for three dialogue corpora – the corpus of dialogues generated with the random policy, the corpus of dialogues generated during the on-line training and the corpus of dialogues generated during the evaluation of the policy trained on the simulated user. The results are given in Table V. Of particular interest is the probability of a user rating the dialogue as successful even though the task was not fully completed. For the random policy corpus, this probability is small, 0.26, whereas for the corpus of dialogues that used the policy trained on the simulated user this probability is very high 0.68. This suggests that when the overall policy behaviour is irrational it is easy for users to identify whether or not the dialogue was successful. However, once the policy behaves more rationally, users find it harder to consistently distinguish between success and failure and they tend to be biased towards success. Similarly, the probability of the user rating the dialogue as successful when the dialogue task actually was fully completed is higher for the trained policies 0.94) than the random policy (0.8). This means that even if the system provides all the required information, but behaves irrationally otherwise, users tend to rate the dialogue as unsuccessful.

In order to further investigate the effect of this inconsistency, 1952 dialogues were used for off-line training during which the system follows the same actions it took in the corpus while re-estimating the policy. In 1362 of these dialogues, the user rating was consistent with the full completion rating. A second policy was then trained off-line on this filtered subset of accurately rated dialogues. The performance of the two policies were then compared using the simulated user, performing 2000 dialogues over a range of semantic error rates from 0 to 50%. The results are shown in Fig. 8 and clearly demonstrate that the accuracy of the reward is crucial for successful on-line learning.

TABLE V
COMPARISON BETWEEN SUBJECTIVE AND OBJECTIVE SCORES FOR DIFFERENT CORPORA

	Random policy	**Online learning**	**Simulator trained**
User feedback	36.3	76.9	85.7
Full completion	17.7	53.8	63.7
$p(\text{feedbck}= 1\|\text{complt}= 1)$	0.80	0.94	0.94
$p(\text{feedbck}= 1\|\text{complt}= 0)$	0.26	0.57	0.68
Total dialogues	114	2960	466

V. CONCLUSIONS

This paper has described a method by which Gaussian process based reinforcement learning can be used to train a dialogue policy from scratch in just a few hundred dialogues without needing a user simulator for bootstrapping and, using a learning strategy that reduces the risk of taking bad actions. The performance of the resulting system was similar to a system trained to convergence on a user simulator, but not significantly better. Given that the user simulator has been

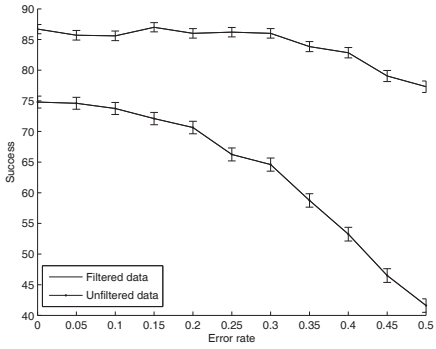

Fig. 8. Performance on simulated user of policies trained offline with filtered and unfiltered dialogues

developed over several man-years of effort, the failure to achieve improved performance may simply be due to the close match between the simulator and the behaviour of real users.

A second contribution of the paper is an investigation of adaptation behaviour when the system is allowed to continue learning for several thousand more dialogues. In this case, some interesting phenomena were observed. In particular, performance did not asymptote towards a maximum but instead appeared to slowly cycle. Further investigation uncovered the rather poor ability of users to give an accurate assessment as to whether or not the dialogue was successful, and hence provide an accurate reward function. Several additional experiments were described which showed that although the POMDP dialogue system is robust to speech understanding errors, the current learning algorithm is not robust to errors in the reward function.

There are two quite different approaches to solving this problem. Firstly, when training in interaction with real users, methods need to be developed for computing a reward signal which do not rely solely on asking the user directly if they were satisfied. Emotion detection, for example, might provide a more appropriate and less intrusive measure and depending on the application, subsequent monitoring of the user's behaviour may give a further indication of success (e.g. continuing on to make a reservation or hanging-up).

Whatever metrics are used for generating reward signals, it is clear that they are likely to be noisy and hence, a second thread of future work needs to focus on robustness to reward signal noise. In theory, reinforcement learning does not require an accurate estimate of the reward given infinite training samples since all that is needed is the expected reward (see Eq. 1). However, for fast policy learning it is necessary that the observations are close to their expected values. If that is not the case, then a reward model is needed which can compensate for noise.

The Gaussian process model for the Q-function takes into account the correlations between different points in the summary state and action spaces and it also assumes that the observations are noisy (Eq. 2). However, the noise in the reward is considered to be static and does not depend on the time when it is observed (Eq. 3). Future work is therefore required to adapt the reward noise estimate during training to make learning robust to the kinds of unexpected phenomena encountered here.

ACKNOWLEDGMENT

The authors would like to thank Rogier van Dalen for valuable suggestions. This work was partially funded by the EPSRC PhD plus programme.

REFERENCES

[1] E. Levin, R. Pieraccini, and W. Eckert, "Using Markov Decision Processes for Learning Dialogue Strategies," in *Proceedings of ICASSP*, 1998.

[2] S. Young, "Talking to Machines (Statistically Speaking)," in *Proceedings of ICSLP*, 2002.

[3] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: MIT Press, 1998.

[4] E. Levin, R. Pieraccini, and W. Eckert, "A Stochastic Model of Human-Machine Interaction for Learning Dialog Strategies," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 11–23, 2000.

[5] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing Dialogue Management with Reinforcement Learning," *Journal of Artificial Intelligence Research*, vol. 16, pp. 105–133, 2002.

[6] N. Roy, J. Pineau, and S. Thrun, "Spoken dialogue management using probabilistic reasoning," in *Proceedings of ACL*, 2000.

[7] B. Zhang, Q. Cai, J. Mao, E. Chang, and B. Guo, "Spoken Dialogue Management as Planning and Acting under Uncertainty," in *Proceedings of Eurospeech*, 2001.

[8] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management," *Computer Speech and Language*, vol. 24, no. 2, pp. 150–174, 2010.

[9] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.

[10] M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, K. Yu, and S. Young, "Training and evaluation of the HIS-POMDP dialogue system in noise," in *Proceedings of SIGDIAL*, 2008.

[11] Y. Engel, S. Mannor, and R. Meir, "Bayes Meets Bellman: The Gaussian Process Approach to Temporal Difference Learning," in *Proceedings of ICML*, 2003.

[12] M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, K. Yu, and S. Young, "Gaussian processes for fast policy optimisation of pomdp-based dialogue managers," in *Proceedings of SIGDIAL*, 2010.

[13] L. Daubigney, M. Gašić, S. Chandramohan, M. Geist, O. Pietquin, and S. Young, "Uncertainty management for on-line optimisation of a POMDP-based large-scale spoken dialogue system," in *Proceedings of Interspeech*, 2011.

[14] F. Jurčíček, S. Keizer, M. Gašić, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk," in *Proceedings of Interspeech*, 2011.

[15] M. Gašić and S. Young, "Effective Handling of Dialogue State in the Hidden Information State POMDP Dialogue Manager," *ACM Transactions on Speech and Language Processing*, 2011.

[16] C. Rasmussen and C. Williams, "Software for Gaussian Processes Regression and Classification," 2007, http://www.gaussianprocess.org/gpml/code/matlab/doc/.

[17] Y. Engel, S. Mannor, and R. Meir, "Reinforcement learning with Gaussian processes," in *Proceedings of ICML*, 2005.

[18] J. Williams, "The best of both worlds: unifying conventional dialog systems and POMDPs," in *Proceedings of Interspeech*, 2008.

[19] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: MIT Press, 2005.

[20] M. Gašić, "Statistical dialogue modelling," Ph.D. dissertation, University of Cambridge, 2011.

[21] Amazon, "Amazon Mechanical Turk," 2011, https://www.mturk.com/mturk/welcome.