Multi-agent learning in multi-domain spoken dialogue systems

M. Gašić, N. Mrkšić, L. Rojas-Barahona, P-H. Su, D. Vandyke, T-H. Wen
Cambridge University Engineering Department
Trumpington St, Cambridge CB2 1PZ, UK
{mg436,nm480,lmr46,phs26,djv27,thw28}@cam.ac.uk

Abstract

The use of a committee of dialogue policies has been shown to be particularly beneficial for adaptation in multi-domain dialogue systems. In this model, each domain is represented by a committee member and the committee collaboratively decides which action the system should take. The action selection is optimised via a reinforcement learning algorithm where the reward is given to the committee member that refers to the current dialogue domain. While such a framework has improved the learning rates compared to single policy models, the main drawback is the need to know which domain the dialogue is in. This is a problem in real-world situations where the domain is a priori unknown, the user request might fall under several domains and the user may switch domains within the same dialogue. In order to deal with this, we augment the policy committee framework using ideas from multi-agent learning. Simulation results and a real user trial show that it is possible to optimise the committee decisions without prior knowledge of the domain.

1 Introduction

Statistical approaches to dialogue management have been shown to reduce design costs and provide superior performance to hand-crafted systems particularly in noisy environments [1]. Traditionally, spoken dialogue systems were built for limited domains described by an underlying *ontology*, which is essentially a structured representation of the database of entities that the dialogue system can talk about.

The semantic web is an effort to organise a large amount of information available on the Internet into a structure that can be more easily processed by a machine designed to perform reasoning about this data [2]. *Knowledge graphs* are one instance of such structures. They typically consist of a set of triples, where each triple represents two entities connected by a specified relationship. Current knowledge graphs have millions of entities and billions of relations and are constantly growing. There has been a significant amount of work in spoken language understanding focused on exploiting knowledge graphs in order to improve coverage [3, 4]. More recently there have been some initial attempts to build statistical dialogue systems that operate on large knowledge graphs, but limited so far to the problem of belief tracking [5]. In this work, we address the problem of decision-making.

Moving from a limited domain dialogue system that operates on a relatively modest ontology to an open domain dialogue system that can converse about anything in a large knowledge graph is a non-trivial problem. An open domain dialogue system can be seen as a (large) set of limited domain dialogue systems. If each of them were trained separately then an operational system would require sufficient training data for every possible topic in the knowledge graph, which is simply not feasible. What is more likely is that instead we have limited and varied data drawn from different domains.

The architecture of a statistical dialogue system typically provides a single policy model that proposes actions throughout the dialogue [1]. This was initially also the case for multi-domain systems [6]. Multi-policy models have been proposed in the context of hierarchical modelling, where the decision-making process follows a hierarchy of policies, but at any given time only one policy makes a decision [7]. Concurrent policy models have been studied in [8] where several policies can propose an action at any given time and heuristics are used to decide which policy should be followed. Combining outputs of multiple policies was previously studied in the context of combining statistical and hand-crafted policies [9, 10]. Previous work on multi-domain dialogue systems has proposed a distributed architecture where a generic policy can be trained on data coming from different domains and later specialised to provide in-domain performance once sufficient data becomes available [11]. A policy committee model has been proposed [12] based on a Bayesian committee machine (BCM) [13]. It consists of a number of policies trained on different, potentially small, datasets. At any given time, when the system needs to make a decision, it consults each committee member and they each propose an action. A data-driven combination method is then used to reach the consensus.

While such a framework has improved the learning rates, the main drawback is that the dialogue manager needs to know which domain the dialogue is in. This is a problem in real world situations where the domain is a priori unknown, the user request might fall under several domains and the user may switch domains within the same dialogue. In order to deal with this, we augment the policy committee framework using ideas from multi-agent learning and show that it is possible to optimise the committee decisions without prior knowledge of the domain.

The rest of the paper is organised as follows. In Sections 2 and 3 we review Gaussian process reinforcement learning and the Bayesian committee machine respectively. Following that, in Section 4, we describe how multi-agent learning can be applied to the policy committee model. Section 5 presents the experimental set-up, followed by an evaluation of several committee combination models using a simulated user (Section 6) and real users (Section 7). We conclude the paper in Section 8 with a summary and future work directions.

2 Gaussian process reinforcement learning

The input to a statistical dialogue manager is typically an N-best list of scored hypotheses obtained from the spoken language understanding unit. Based on this input, at every dialogue turn, a distribution of possible dialogue states called the *belief state*, an element of *belief space* $\boldsymbol{b} \in \mathcal{B}$, is estimated. The quality of a dialogue is defined by a *reward function* $r(\boldsymbol{b}, a)$ and the role of a dialogue policy π is to map the belief state $\boldsymbol{b}$ into a system action, an element of *action space* $a \in \mathcal{A}$, at each turn so as to maximise the expected cumulative reward.

The expected cumulative reward for a given belief state $\boldsymbol{b}$ and action a is defined by the Q-function:

$$Q(\boldsymbol{b}, a) = E_\pi \left(\sum_{\tau=t+1}^{T} \gamma^{\tau-t-1} r_\tau | b_t = \boldsymbol{b}, a_t = a \right) \tag{1}$$

where r_τ is the immediate reward obtained at time τ, T is the dialogue length and γ is a discount factor, $0 < \gamma \leq 1$. Optimising the Q-function is then equivalent to optimising the policy π.

GP-Sarsa is an on-line reinforcement learning algorithm that models the Q-function as a Gaussian process [14]:

$$Q(\boldsymbol{b}, a) \sim \mathcal{GP}\left(0, k((\boldsymbol{b}, a), (\boldsymbol{b}, a))\right) \tag{2}$$

where the kernel $k(\cdot, \cdot)$ is factored into separate kernels over belief and action spaces $k_{\mathcal{B}}(\boldsymbol{b}, \boldsymbol{b}')k_{\mathcal{A}}(a, a')$.

For a training sequence of belief state-action pairs $\boldsymbol{B} = [(\boldsymbol{b}^0, a^0), \ldots, (\boldsymbol{b}^t, a^t)]^\mathsf{T}$ and the corresponding observed immediate rewards $\boldsymbol{r} = [r^1, \ldots, r^t]^\mathsf{T}$, the posterior of the Q-function for any belief state-action pair $(\boldsymbol{b}, a)$ is given by:

$$Q(\boldsymbol{b}, a) | \boldsymbol{r}, \boldsymbol{B} \sim \mathcal{N}(\overline{Q}(\boldsymbol{b}, a), cov((\boldsymbol{b}, a), (\boldsymbol{b}, a))) \tag{3}$$

where the posterior mean and covariance take the form:

$$\begin{aligned}\overline{Q}(\boldsymbol{b},a) &= \boldsymbol{k}(\boldsymbol{b},a)^{\mathsf{T}}\boldsymbol{H}^{\mathsf{T}}(\boldsymbol{H}\boldsymbol{K}\boldsymbol{H}^{\mathsf{T}}+\sigma^2\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}})^{-1}\boldsymbol{r},\\ cov((\boldsymbol{b},a),(\boldsymbol{b},a)) &= k((\boldsymbol{b},a),(\boldsymbol{b},a))-\\ &\quad \boldsymbol{k}(\boldsymbol{b},a)^{\mathsf{T}}\boldsymbol{H}^{\mathsf{T}}(\boldsymbol{H}\boldsymbol{K}\boldsymbol{H}^{\mathsf{T}}+\sigma^2\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}})^{-1}\boldsymbol{H}\boldsymbol{k}(\boldsymbol{b},a)\end{aligned} \tag{4}$$

where $\boldsymbol{k}(\boldsymbol{b},a) = [k((\boldsymbol{b}^0,a^0),(\boldsymbol{b},a)),\ldots,k((\boldsymbol{b}^t,a^t),(\boldsymbol{b},a))]^{\mathsf{T}}$, $\boldsymbol{K}$ is the Gram matrix [15], $\boldsymbol{H}$ is a band matrix with diagonal $[1,-\gamma]$ and σ^2 is an additive noise factor which controls how much variability in the Q-function estimate is expected during the learning process. Since the Gaussian process for the Q-function defines a Gaussian distribution for every belief state-action pair (3), when a new belief point $\mathbf{b}$ is encountered, for each action $a \in \mathcal{A}$, there is a Gaussian distribution over Q-values. Sampling from these Gaussian distributions gives Q-values $\hat{Q}(\mathbf{b},a) \sim \mathcal{N}(\overline{Q}(\mathbf{b},a),\Sigma^Q(\boldsymbol{b},a))$ where $\Sigma^Q(\boldsymbol{b},a) = cov((\boldsymbol{b},a),(\boldsymbol{b},a))$ from which the action with the highest sampled Q-value can be selected:

$$\pi(\mathbf{b}) = \arg\max_a \left\{\hat{Q}(\mathbf{b},a) : a \in \mathcal{A}\right\}. \tag{5}$$

To use GPRL for dialogue, a kernel function must be defined on both the belief state space $\mathcal{B}$ and the action space $\mathcal{A}$. Here we use the Bayesian Update of Dialogue State (BUDS) dialogue model [16]. The action space consists of a set of slot-dependent and slot-independent summary actions which are mapped to master actions using a set of rules and the kernel is defined as:

$$k_{\mathcal{A}}(a,a') = \delta_a(a') \tag{6}$$

where $\delta_a(a') = 1$ iff $a = a'$, 0 otherwise. The belief state consists of the probability distributions over the Bayesian network hidden nodes that relate to the dialogue history for each slot and the user goal for each slot. The dialogue history nodes can take a fixed number of values, whereas user goals range over the values defined for that particular slot in the ontology and can have very high cardinalities. User goal distributions are therefore sorted according to the probability assigned to each value since the choice of summary action does not depend on the values but rather on the overall shape of each distribution.

The kernel function over both dialogue history and user goal nodes is based on the expected likelihood kernel [17], which is a simple linear inner product. The kernel function for belief space is then the sum over all the individual hidden node kernels:

$$k_{\mathcal{B}}(\boldsymbol{b},\boldsymbol{b}') = \sum_h \langle \boldsymbol{b}_h, \boldsymbol{b}'_h \rangle \tag{7}$$

where $\boldsymbol{b}_h$ is the probability distribution encoded in the h^{th} hidden node.

3 Bayesian committee machine

The Bayesian committee machine is an approach to combining estimators that have been trained on different datasets and can be applied to Gaussian process regression [13]. Here we apply this method to combine the outputs of multiple estimates of Q-values Q_i with mean $\overline{Q}_i$ and covariance Σ_i^Q, given by Eq. 4 and trained on a set of rewards and belief-state action pairs $\boldsymbol{r}_i, \boldsymbol{B}_i$ for $i \in \{1,\ldots,M\}$, where M is the number of policies in the policy committee.

Following the description in [13], the combined mean $\overline{Q}$ and covariance Σ^Q are calculated as:

$$\begin{aligned}\overline{Q}(\boldsymbol{b},a) &= \Sigma^Q(\boldsymbol{b},a)\textstyle\sum_{i=1}^M \Sigma_i^Q(\boldsymbol{b},a)^{-1}\overline{Q}_i(\boldsymbol{b},a),\\ \Sigma^Q(\boldsymbol{b},a)^{-1} &= -(M-1)*k((\boldsymbol{b},a),(\boldsymbol{b},a))^{-1} + \textstyle\sum_{i=1}^M \Sigma_i^Q(\boldsymbol{b},a)^{-1}.\end{aligned} \tag{8}$$

4 Multi-agent learning in the policy committee framework

In the standard reinforcement learning framework there is a single agent that is trying to solve one task in the given environment. However, for complex problems it has been shown [18] that

it is more effective to decompose the problem into subproblems and have a system with multiple modules where each module is trying to solve a subtask. Therefore, each module takes into account only a part of the state space. This can significantly speed up the learning process. The learning in a multi-agent system is typically performed in thee steps [18]. First, each agent proposes an action. Second, a gating mechanism is deployed which selects the resulting system action. This mechanism can be either handcrafted or optimised automatically. Finally, the reward that the system gets is distributed among the agents and they each re-estimate their policy.

As can be seen, the multi-agent framework is closely related to the policy committee model. In fact, the first two steps are exactly the same: each committee member estimates its own Q-function and then Eq. 8 is used as the gating to automatically combine the output. In this work, we include the third step, which is distributing the reward so that each committee member can learn from every dialogue. So far, the total reward was given only to the committee member that is estimating the Q-function for the current dialogue domain [12]. In practice it may be difficult to know which domain the dialogue is currently in, the same user request may relate to different domains and the user can switch domains within the same dialogue. Therefore we adopt three strategies for distributing the reward:

naïve approach: the total reward that the system obtains is directly fed back to each committee member [18]

winner-takes-all approach: the total reward that the system obtains is fed back to the committee member that proposed the highest Q-value for the action that was finally chosen by the gating mechanism [19]

reward scaling approach: the total reward is redistributed to each committee member in such way to reflect its contribution to the final action chosen by the gating mechanism [18]

5 Experimental set-up

In order to examine the ability of the proposed method to operate on multiple domains we examine three domains: SFR consisting of restaurants in San Francisco, SFH consisting of hotels in San Francisco, L11 consisting of laptops with 11 properties that the user can specify. In that way not only the slots are different, but one of the domains, L11, has many more slots than others. A description of each domain along with slots is given in [12].

6 Simulation results

In order to investigate the effectiveness of multi-agent learning within the policy committee model, a variety of contrasts were examined using an agenda-based simulated user operating at the dialogue act level [20]. The reward function allocates -1 at each turn to encourage shorter dialogues, plus 20 at the end of each successful dialogue. The user simulator includes an error generator and this was set to generate incorrect user inputs 15% of time.

The contrasts studied were as follows:

NAÏV **Naïve approach** – The total reward was given to each committee member.

WINN **Winner-takes-all approach** – The total reward was given to the policy member which on average was giving the highest Q-value Q-variance ratio, $\Sigma_i^Q(\boldsymbol{b}, a)^{-1}\overline{Q}_i(\boldsymbol{b}, a)$ from Eq. 8, for the action that was taken by the system.

SELFSCALE **Scale received reward according to self Q-value estimate** – Each policy committee member scales the total reward according to the average portion of its associated Q-value for the action that the system took as opposed to other actions it proposed.

COMMSCALE **Scale received reward according to all committee members' Q-value estimate** – Each policy committee member scales the total reward according to its average portion of the Q-value Q-variance ratio, $\Sigma_i^Q(\boldsymbol{b}, a)^{-1}\overline{Q}_i(\boldsymbol{b}, a)$ from Eq. 8, for the action that the system took as opposed to the Q-value Q-variance ratio that other committee members proposed for the taken action.

Table 1: Comparison of strategies for multi-domain adaptation. In-domain performance is measured in terms of reward, success rate and the average number of turns per dialogue. Results are given with 95% confidence intervals.

Strategy	Reward	Success	#Turns
SFR trained on 750 dialogues from SFR, SFH, L11			
NAÏV	7.00 ± 0.20	73.66 ± 0.86	7.70 ± 0.08
WINN	6.84 ± 0.21	75.81 ± 0.84	8.29 ± 0.09
SELFSCALE	6.86 ± 0.20	72.90 ± 0.87	7.68 ± 0.08
COMMSCALE	7.06 ± 0.21	75.29 ± 0.85	7.98 ± 0.09
MBCM	7.37 ± 0.20	76.60 ± 0.83	7.92 ± 0.08
L11 trained on 750 dialogues from SFR, SFH, L11			
NAÏV	8.82 ± 0.20	77.40 ± 0.82	6.63 ± 0.07
WINN	7.23 ± 0.22	72.35 ± 0.88	7.20 ± 0.09
SELFSCALE	7.69 ± 0.21	72.17 ± 0.88	6.72 ± 0.07
COMMSCALE	8.11 ± 0.21	74.61 ± 0.85	6.78 ± 0.08
MBCM	8.52 ± 0.20	77.09 ± 0.82	6.88 ± 0.07
SFR trained on 7500 dialogues from SFR, SFH, L11			
NAÏV	9.45 ± 0.22	87.98 ± 0.85	8.14 ± 0.11
WINN	9.67 ± 0.18	89.24 ± 0.68	8.15 ± 0.09
COMMSCALE	9.41 ± 0.17	88.08 ± 0.66	8.18 ± 0.09
MBCM	9.67 ± 0.17	88.28 ± 0.66	7.96 ± 0.08
L11 trained on 7500 dialogues from SFR, SFH, L11			
NAÏV	10.92 ± 0.16	86.80 ± 0.70	6.42 ± 0.07
WINN	11.25 ± 0.18	88.51 ± 0.76	6.43 ± 0.08
COMMSCALE	11.24 ± 0.17	88.55 ± 0.69	6.44 ± 0.07
MBCM	10.73 ± 0.16	87.23 ± 0.66	6.70 ± 0.07
Averaged across domains and size of training data			
NAÏV	8.94 ± 0.10	80.49 ± 0.42	7.13 ± 0.04
WINN	8.46 ± 0.10	80.37 ± 0.42	7.58 ± 0.05
COMMSCALE	8.83 ± 0.10	81.17 ± 0.40	7.38 ± 0.04
MBCM	9.06 ± 0.09	82.17 ± 0.38	7.35 ± 0.04

MBCM **Multi-policy Bayesian committee machine** – Each committee member is trained only on in-domain data, so the reward is passed only to the committee member which is dedicated for that domain. This method was introduced in [12] and requires the knowledge of the domain.

We examine two cases: when the training data is limited, with only 250 dialogues available for each domain, and when there is more training data available, 2500 for each domain. Similar to [12], we consider a multi-domain system for SFR, SFH and L11.

For each method described above, 10 policies were trained on the simulated user using different random seeds. Each policy was then evaluated using 1000 dialogues on each domain. The overall average reward, success rate and number of turns are given in Table 1 together with 95% confidence intervals.

There are few important conclusions to be drawn from the results. First, on a smaller dataset the approach which chooses the winner committee member to pass the total reward to is less effective than the approaches which distribute the reward. This is expected, as in the latter case the policy learns from a larger poll of dialogues, which is particularly useful in early stages of the optimisation process. Out of all methods that distribute the reward, the method which uses self-scaling has the poorest performance, so its performance is not examined on a larger dataset. On larger datasets, the winner-takes-all approach has better or equal performance to the approaches which distribute the reward. This is in line with the intuition that with abundance of data, the accuracy of the given reward is more important than the size of training data. If we average results across the domains and the sizes of the training data, we can see that it is still more effective to use the approaches which distribute the reward.

7 Real user evaluation

To fully examine the effectiveness of the proposed reward distribution approach, two set-ups were also trained in direct interaction with human users. First, a multi-policy Bayesian committee machine (MBCM) is trained from scratch using data from the SFR, SFH and L6[1] domains, as presented in [11]. This policy committee operates on all three domains but is dependent on the knowledge of the current domain for policy updating. We compare this to the committee reward scaling (COMMSCALE), presented in Section 6, which distributes the reward to each committee member for each dialogue. We deployed the system in a telephone-based set-up, with subjects recruited via Amazon MTurk, same as in [11]. A recurrent neural network model was used to estimate the reward [21].

Fig. 1 shows the moving average reward as a function of the number of training dialogues for the L6 domain comparing the MBCM and COMMSCALE committee approaches. The committees were also trained on SFR and SFH domains in parallel. The training data across the domains was equally distributed. The moving window was set to 100 dialogues so that after the initial 100 dialogues each point on the graph is an average of 300 dialogues. The shaded area represents a 95% confidence interval. As can be seen, the results confirm that it is not necessary for the committee to be aware of the domain, on the contrary, distributing reward to each committee member according to their contribution can even produce better performance than only sending the reward signal to the committee member dedicated to the current domain.

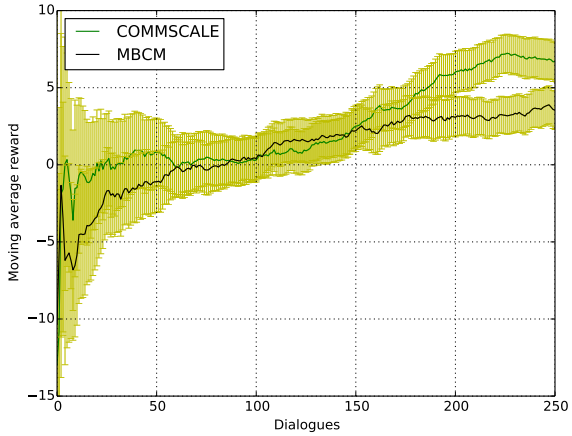

Figure 1: Training in interaction with human users on L6 domain – moving average reward

8 Conclusions and Future work

We have extended the policy committee model using ideas from multi-agent learning to distribute the reward signal among the committee members, making this model particularly useful in a real-world scenario where the domain is a priori unknown. In simulations, the proposed approach achieves a performance close to that which relies on domain information to assign the reward to the appropriate committee member, while in a real human trial, it produced better performance.

In the work presented here, it is only the action-selection component that does not use any domain information. Choosing which domain information to add to the chosen action, as well as the spoken language understanding, requires the knowledge of the domain. In future, we plan to apply this method in combination with a domain or a topic tracker operating over a large knowledge graph, with the aim of building open domain statistical dialogue systems.

Acknowledgments

This research was funded by the EPSRC grant EP/M018946/1 *Open Domain Statistical Spoken Dialogue Systems*, data is available at https://www.repository.cam.ac.uk/handle/1810/252636.

[1]L6 is a laptop domain which contains 6 attributes that the user can specify.

References

[1] SJ Young, M Gašić, B Thomson, and JD Williams, "Pomdp-based statistical spoken dialogue systems: a review," *Proceedings IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[2] Péter Szeredi, Gergely Lukácsy, and Tamás Benkö, *The Semantic Web Explained: The Technology and Mathematics Behind Web 3.0*, Cambridge University Press, New York, NY, USA, 2014.

[3] Gökhan Tür, Minwoo Jeong, Ye-Yi Wang, Dilek Hakkani-Tür, and Larry P Heck, "Exploiting the semantic web for unsupervised natural language semantic parsing," in *Proceedings of Interspeech*, 2012.

[4] Larry P Heck, Dilek Hakkani-Tür, and Gökhan Tür, "Leveraging knowledge graphs for web-scale unsupervised semantic parsing.," in *Proceedings of Interspeech*, 2013, pp. 1594–1598.

[5] Yi Ma, Paul A. Crook, Ruhi Sarikaya, and Eric Fosler-Lussier, "Knowledge graph inference for spoken dialog systems," in *Proceedings of 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015*. April 2015, IEEE Institute of Electrical and Electronics Engineers.

[6] Z Wang, H. Cheng, G. Wang, H. Tian, H. Wu, and H. Wang, "Policy learning for domain selection in an extensible multi-domain spoken dialogue system," in *EMNLP*, 2014.

[7] Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira, "Evaluation of a hierarchical reinforcement learning spoken dialogue system," *Comput. Speech Lang.*, vol. 24, no. 2, pp. 395–429, Apr. 2010.

[8] Pierre Lison, "Multi-policy dialogue management," in *Proceedings of the SIGDIAL 2011 Conference*, Stroudsburg, PA, USA, 2011, SIGDIAL '11, pp. 294–300, Association for Computational Linguistics.

[9] Jason D. Williams, "The best of both worlds: Unifying conventional dialog systems and pomdps," in *Proc Interspeech, Brisbane, Australia*, 2008.

[10] M. Gašić, F. Lefevre, F. Jurcicek, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Back-off action selection in summary space-based pomdp dialogue systems," in *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, Nov 2009, pp. 456–461.

[11] M. Gašić, D. Kim, P. Tsiakoulis, and S. Young, "Distributed dialogue policies for multi-domain statistical dialogue management," in *Proceedings of ICASSP*, 2015.

[12] M. Gašić, N. Mrkšić, P-H. Su, D. Vandyke, T-H. Wen, and S. Young, "Policy committee for adaptation in multi-domain spoken dialogue systems," in *Proceedings of ASRU*, 2015.

[13] Volker Tresp, "A Bayesian Committee Machine," *Neural Comput.*, vol. 12, no. 11, pp. 2719–2741, Nov. 2000.

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

[15] CE Rasmussen and CKI Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Massachusetts, 2005.

[16] 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.

[17] T Jebara, R Kondor, and A Howard, "Probability product kernels," *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, Dec. 2004.

[18] Peter Raicevic, "Parallel reinforcement learning using multiple reward signals," *Neurocomputing*, vol. 69, no. 1618, pp. 2171 – 2179, 2006.

[19] Mark Humphrys, "W-learning: competition among selfish Q-learners," Tech. Rep. UCAM-CL-TR-362, University of Cambridge, Computer Laboratory, Apr. 1995.

[20] J Schatzmann, B Thomson, K Weilhammer, H Ye, and SJ Young, "Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System," in *Proceedings of HLT*, 2007.

[21] Pei-Hao Su, David Vandyke, Milica Gašić, Dongho Kim, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young, "Learning from real users: Rating dialogue success with neural networks for reinforcement learning in spoken dialogue systems," in *Proceedings of Interspeech*, 2015.