Inverse Reinforcement Learning for Micro-Turn Management

Dongho Kim, Catherine Breslin, Pirros Tsiakoulis, Milica Gašić, Matthew Henderson, Steve Young

Department of Engineering, University of Cambridge, Cambridge, UK

{dk449,cb404,pt344,mg436,mh521,sjy}@cam.ac.uk

Abstract

Existing spoken dialogue systems are typically not designed to provide natural interaction since they impose a strict turn-taking regime in which a dialogue consists of interleaved system and user turns. To allow more responsive and natural interaction, this paper describes a system in which turn-taking decisions are taken at a more fine-grained micro-turn level. A decision-theoretic approach is then applied to optimise turn-taking control. Inverse reinforcement learning is used to capture the complex but natural behaviours from human-human dialogues and optimise interaction without specifying a reward function manually. Using a corpus of human-human interaction, experiments show that IRL is able to learn an effective reward function which outperforms a comparable handcrafted policy.

Index Terms: dialogue management, spoken dialogue systems, inverse reinforcement learning, Markov decision processes

1. Introduction

A major shortcoming of traditional spoken dialogue systems is that they are not able to provide natural interaction since they impose a strict turn-taking regime in which a dialogue consists of interleaved system and user turns. For example, when the user speaks, the system must listen until a silence detector determines that the user has finished. When the user stops speaking, the system then takes the floor, responds, and hands the floor back to the user. The only flexibility offered is that most systems allow the user to barge in on the system. This rigid turn-taking is not only unnatural, it is also difficult to implement because silence detectors have difficulty distinguishing between silence and background noise. It also introduces unwanted latency since the system must wait to ensure that the silence really does mark the end of the user turn and is not just a short pause in the middle of the utterance.

To allow more responsive and natural interaction, a variety of decision-theoretic approaches have been proposed, which provide a more principled way of optimising the control of turn-taking [1, 2, 3, 4, 5]. Since any turn-taking decision may have an effect on the future evolution of the dialogue, a general solution should view the problem as sequential decision making in which the system has to optimise a series of turn-taking actions over a dialogue, such as listening to the user, speaking, or interrupting. Furthermore, by taking these actions at a more fine-grained micro-turn level, *e.g.*, every 100ms, micro-turn management can offer richer interaction which can incorporate natural discourse phenomena such as barge-in and backchannels. This optimisation is guided by the computation of the expected utilities of different micro-turn actions given a cost or reward function.

Existing attempts to apply and optimise turn-taking, typically either use a handcrafted decision policy or a manually specified reward function based on the simple assumption that conversants attempt to minimise gaps and overlaps. However, unlike higher level of dialogue management [6], it is unclear how a reward function can be specified which is able to capture the complex but natural behaviours required for natural conversation.

In this paper, we present an application of Inverse Reinforcement Learning (IRL) [7] using the Markov Decision process (MDP) formalism [8] applied to micro-turn interaction. This allows the reward function used by human conversants to be automatically recovered by observing human-human interactions and thus flexibly model the many discourse phenomena encountered in natural dialogues.

2. IRL for micro-turn management

Traditional approaches to learning decisions based on demonstrations of the required behaviours by an *expert* typically use general-purpose supervised learning methods, which *directly* learn the policy as a mapping from states to actions [9]. However, such approaches fail to learn good policies in those parts of the environment which the expert tends to avoid, since training samples are then very sparse in these regions [10].

IRL is a more recent approach which frames the learnt policies as solutions of MDPs. The key assumption here is that experts behave near-optimally to maximise rewards along the sequential decision process, and we must find an unknown reward function that makes their demonstrated behaviour appear near-optimal. The experts' policy can be recovered *indirectly* by solving the MDP with the learnt reward function. Hence, the problem is reduced to a task of recovering a reward function that induces the demonstrated behaviour. This approach allows the experts' policy to be generalised to unobserved situations. In addition, the learnt reward function is transferable to different tasks.

2.1. IRL preliminaries

An MDP is defined as a tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathbf{r}, \gamma\}$: $\mathcal{S}$ is the set of states s; $\mathcal{A}$ is the set of actions a; $\mathcal{T}$ is the transition function where $\mathcal{T}_{s'}^{sa}$ denotes the probability $P(s'|s,a)$ of reaching state s' from state s by taking action a; $\mathbf{r}$ is the reward function where $\mathbf{r}_{sa}$ denotes the immediate reward $R(s,a)$ of executing action a in state s; $\gamma \in [0,1)$ is the discount factor. The optimal policy π^* maximises the expected discounted sum of rewards $E[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_{s_t a_t} | \pi^*]$. Given $\mathcal{M}\backslash\mathbf{r}$, IRL aims to find a reward function $\mathbf{r}$ under which π^* matches the demonstrated trajectories $\mathcal{D} = \{\zeta_1, \ldots, \zeta_N\}$ where ζ_i is a sequence of state-action pairs $\zeta_i = \{(s_{i,0}, a_{i,0}), \ldots, (s_{i,T_i}, a_{i,T_i})\}$.

We could assume that the examples $\mathcal{D}$ are drawn from the optimal policy π^*. In practice, however, this assumption must be relaxed since human demonstrations can be suboptimal and contain inherent stochasticity. Hence, recent IRL algorithms [11, 12] assume that $\mathcal{D}$ is drawn from a maximum-entropy randomised policy π, in which the probability of executing an action a in state s is proportional to the exponential of the expected total reward after taking the action, denoted $\pi(a|s) \propto \exp(\mathbf{Q}^{\mathbf{r}}_{sa})$, where Q-value $\mathbf{Q}^{\mathbf{r}}$ is defined as:

$$\mathbf{Q}^{\mathbf{r}} = \mathbf{r} + \gamma \mathcal{T} \mathbf{V}^{\mathbf{r}} \ . \tag{1}$$

In the above equation, the value function $\mathbf{V}^{\mathbf{r}}$ is defined as $\mathbf{V}^{\mathbf{r}}_s = \log \sum_a \exp \mathbf{Q}^{\mathbf{r}}_{sa}$, which uses a soft version of the Bellman backup operator $V^{\mathbf{r}}_s = \max_a \mathbf{Q}_{sa}$. The IRL log-likelihood of data given randomised policy π and reward function $\mathbf{r}$ can be written as $\log P(\mathcal{D}|\mathbf{r}) = \sum_i \sum_t \log \pi(a_{i,t}|s_{i,t})$. Note that this randomised policy can be suboptimal compared to the optimal deterministic policy resulting from the Bellman iteration, but may produce more human-like behaviours in micro-turn interaction.

3. Experimental setup

The ultimate goal of this work is to develope an IRL-trained micro-turn manager for a POMDP-based dialogue system providing restaurant information [13]. However, in order to learn natural turn-taking behaviours from demonstrations, we need human-human dialogues in which one speaker represents the dialogue system and and the other represents the user. Since no appropriate human-human data was available in the restaurant domain, dialogue data from a related domain was used with the ultimate aim of transferring the IRL results back to the target domain. In this paper, however, we focus on assessing the advantages of IRL, and defer adaptation to our restaurant system to future work.

3.1. Dataset

To assess the effectiveness of IRL for learning reward functions at the micro turn level, dialogue data from the SpaceBook project[14] was used. This corpus was collected to support the development of a spoken dialogue-based information system for pedestrian navigation and exploration. This data contains human-human dialogues with natural turn-taking and many discourse phenomena including backchannels, barge-ins and overlap in a similar dialogue domain.

One dialogue consists of two audio streams and transcripts: a tourist who plays the role of a user, and a wizard who plays the dialogue system and gives direction to the tourist. These dialogues were aligned to get the start and end times for both words and pauses. Each dialogue was split into 100ms micro-turns and each segment tagged with the corresponding values of state variables which will be explained in the following subsection. The data set consisted of 11 dialogues of approximate duration 2 hours with 33303 micro-turns in total. The length of each dialogue varies from less than 1 minute to 30 minutes.

3.2. Model formulation

Although syntax, semantics and dialogue context can play a vital role in micro-turn interaction, we limit our attention to relatively primitive features such as timing and prosody of utterances and backchannels. In order to utilise higher-level information about users' intentions while keeping the model simple, an end-of-utterance (EOU) classifier distinguishing within-turn pauses from end-of-turn pauses is trained. This uses a set of features including prosody and timing features as proposed in [2]. Three different classifiers were trained on all pauses, pauses longer than 100ms, and pauses longer than 200ms. Hence, each segment in the datasets is tagged with an EOU probability according to the duration-so-far of its corresponding pause. The segment is tagged with 0 probability if the user is not silent throughout it.

We designed a micro-turn MDP which extends the 6-state model proposed in [3]. Information which is intuitively likely to correlate with the turn-taking behaviour can be encoded in the state space $\mathcal{S}$. In this paper, the micro-turn state is defined as $s = \langle \mathtt{m}, \mathtt{u}, \mathtt{l}, \mathtt{b}, \mathtt{e}, \mathtt{c} \rangle$: m and u are the states which indicate the current voice activity by the system and user, respectively. Typical voice activity states are speaking, silence, or generating backchannels; l denotes the identity of the last speaker which is helpful in distinguishing pauses between switching from system to user and pauses between switching from user to system; b denotes the number of backchannels so far in the current system utterance; e denotes the aforementioned EOU probability; c is a timing state variable which denotes the duration-so-far (upto one second) of the current voice activity state of the system or user. e and c were quantised into ten discrete levels. By making some conditional independence assumptions according to the definitions of state variables, the transition probability can also be factored as:

$$\begin{aligned} P(s'|s,a) =& P(\mathtt{m}'|a)P(\mathtt{l}'|\mathtt{l},\mathtt{m},\mathtt{m}',\mathtt{u},\mathtt{u}')P(\mathtt{b}'|\mathtt{b},\mathtt{m},\mathtt{m}',\mathtt{u},\mathtt{u}') \\ & \cdot P(\mathtt{c}'|\mathtt{c},\mathtt{m},\mathtt{m}',\mathtt{u},\mathtt{u}') \qquad (2) \\ & \cdot P(\mathtt{u}'|\mathtt{m},\mathtt{u},\mathtt{l})P(\mathtt{e}'|\mathtt{e},\mathtt{u}') \ . \qquad (3) \end{aligned}$$

Note that the state transitions for variables in (2) are deterministic. However, the user state u and EOU probability e can change probabilistically along the process according to (3). We assume that the user's behaviour depends on the previous voice activity state and the last speaker. We also assume that the EOU probability depends on the previous EOU probability and the current user state. These transition probabilities were empirically estimated from the data.

The micro-turn MDP has three actions: *speak*, *silent* and *backchannel*. The system can grab the floor by executing the *speak* action and then continue speaking. When the user is speaking, the *speak* action entails barging in on the user. The system can release the floor by executing the *silent* action, or generate a backchannel with the *backchannel* action. Note that we assume that a decision epoch may not occur at some micro-turns, as in continuous-time MDPs [8]. Once the system grabs the floor by executing the *speak* action when the user is silent, it does not need to release the floor until the system completes its utterance or the user interrupts. Likewise, when initiating a backchannel, the system does not need to invoke a second action to terminate it since most backchannels are short vocalised sounds or simple words or phrases.

3.3. IRL algorithm

In this work, the Gaussian Process IRL (GPIRL) algorithm [12] was used. While most prior IRL algorithms assume the reward to be a linear combination of a set of features, GPIRL uses GP regression to learn the reward as a nonlinear function. Hence, GPIRL enables us to automatically capture the complex reward structure in micro-turn interaction without enumerating all of the possibly relevant features which are usually logical conjunctions of state variables [15]. Using the automatic relevance detection kernel, GPIRL can also determine the relevance of each feature to the underlying true reward while encouraging a simple reward structure with sparse feature weights.

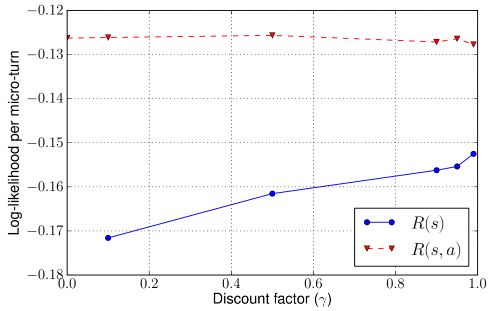


Figure 1: *Training log-likelihood for each discount factor.*

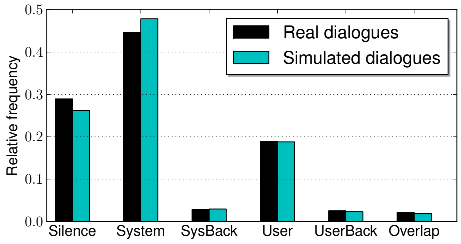


Figure 2: *Distribution over system-user states.*

4. Experimental results

The proposed IRL approach can be compared with a supervised classifier which directly learns a stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$. We note that the IRL approach with zero discount factor ($\gamma = 0$) can be viewed as a simple supervised classifier, as shown in (1). Hence, we compared the different reward functions and policies from the micro-turn MDP with discount factors ranging from 0 to 0.99.

If the underlying true reward function is known, we can directly evaluate the performance by measuring how suboptimal the policy learnt via IRL is under the true reward. However, the true reward is unknown in practice and thus it is non-trivial to evaluate the reward functions. In this paper, we compared the algorithms by measuring log-likelihood to see how well each can model the given data (Section 4.1) and their generalisation capabilities to unseen data (Section 4.2). The latter is especially important in practice where the data is highly sparse: only 883 states were observed in the SpaceBook dataset, out of 3960 possible states. We also analyse the learnt policy to assess if the IRL technique can reproduce human-like behaviours (Section 4.3).

4.1. IRL training results

We tested two forms of reward function R: $R(s)$ which gives the immediate reward of reaching state s and $R(s,a)$ which depends on the state-action pair. By using $R(s)$, we can reduce the number of unknown parameters which will be learnt.

Figure 1 shows training log-likelihood on 11 dialogues over different discount factors. Note that the result of $R(s)$ with $\gamma = 0$ is omitted since it had a poor log-likelihood of -1.14. This is because when $\gamma = 0$, the Q-value defined in (1) is written as $\mathbf{Q}^{\mathbf{r}}_{sa} = \mathbf{r}_s$, and thus the resulting policy will have the same Q-value (*i.e.*, same probability) for every action. Using larger γ, the deeper lookahead search can be performed to take future effects into account, and thus the training log-likelihood of $R(s)$ with larger γ was higher. When we used $R(s,a)$, the performance was clearly better than $R(s)$ since $R(s,a)$ could utilise the actions performed in the given dataset as an additional feature. However, in the case of $R(s,a)$ the log-likelihood in training is not sensitive to the discount factor, and indeed falls as γ increases. This can also be explained by considering $\gamma = 0$. In this case, the IRL algorithm does not perform any lookahead search in (1), and the rewards will be exactly the same as the Q-values, *i.e.*, $\mathbf{Q}^{\mathbf{r}}_{sa} = \mathbf{r}_{sa}$. However, it is still allowed to directly search for the good rewards which is similar to Q-values from IRL with lookahead search ($\gamma > 0$). Hence, the resulting rewards vary across different γ, but Q-values remain similar. Although it may seem that IRL provides no benefit over supervised classification, it does in fact generalise better as will be demonstrated in the next subsection.

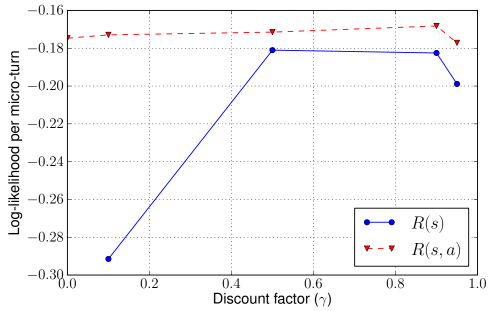


Figure 3: *Testing log-likelihood for each discount factor.*

Figure 2 shows the relative frequency of the voice activity state of the machine and user in real and simulated dialogues. Among 9 possible combinations, the most common 5 states (silence; the system or the user is either speaking or generating a backchannel while the opponent is silent) and overlap state (both the system and user is speaking or generating backchannels) are shown. As can be seen, the IRL policy successfully reproduced the statistical properties of the real dialogues.

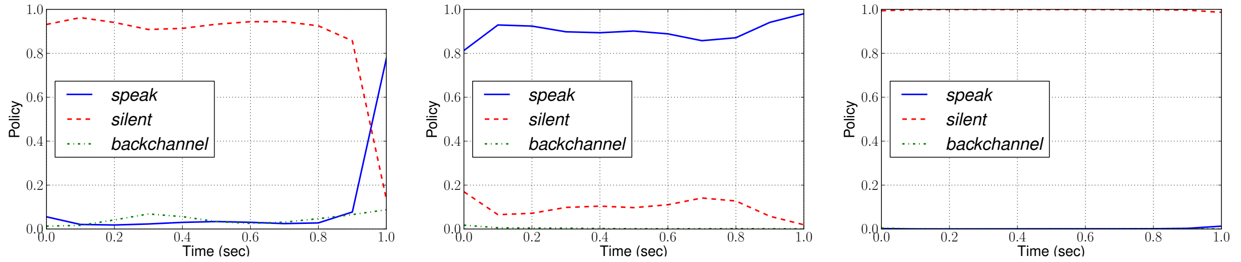
4.2. Cross-validation

In order to demonstrate the generalisation capability of the IRL technique, we conducted cross-validation experiments. The reward function was trained on one dialogue and evaluated on the remaining 10 dialogues. The environment $\mathcal{T}$ in a training dialogue might be different from that of the test dialogues, and thus we computed a maximum-entropy randomised policy in the test environments given the learnt reward function by solving the micro-turn MDP. We then tested the policy. We excluded one dialogue shorter than one minute, which was too sparse to learn the reward function. Therefore the following results were averaged over 10 experiments.

Figure 3 shows the averaged log-likelihood on the test sets. For $R(s,a)$, there is a clear trend of increasing log-likelihood as γ increases upto 0.9, whereas the log-likelihood for $R(s)$ is maximum at the slightly lower value of 0.8, perhaps because of the reduced lookahead capability. Compared to the results in Figure 1, it is clear that IRL with an appropriate discount factor generalises better than supervised classification approaches. However, it is interesting to note that the performance of $\gamma \geq 0.95$ was degraded in both cases. We suspect that using IRL with a deep lookahead search makes it very susceptible to errors in the estimated environmental model $\mathcal{T}$ which is inherently noisy due to the limited amount of training data.

4.3. Policy assessment

We now illustrate with specific examples how the policy learnt via IRL makes micro-turn decisions. Figure 4a shows the IRL



(a) *Silence.*

(b) *Overlap.*

(c) *Listening.*

Figure 4: *Illustrative examples of IRL policy. The horizontal axis in each subfigure represents elapsed time since the state changed to (a) "silence" in which both keep silent (b) "overlap" in which both are speaking (c) "listening" in which the system is listening to the user. The vertical axis represents the probability of each micro-turn action.*

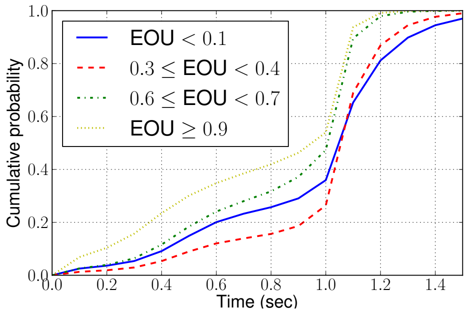


Figure 5: *Effect of End-of-utterance (EOU) probability. The horizontal axis represents elapsed time since the state changed to "silence" in which both keep silent. The vertical axis represents the cumulative probability of the system executing a speak or backchannel action.*

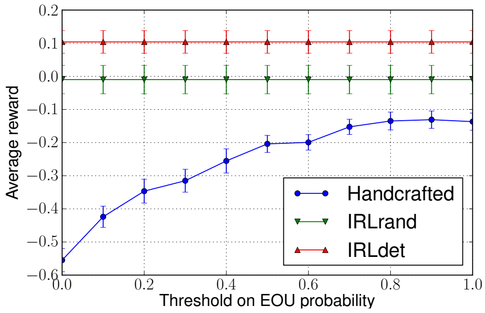


Figure 6: *Average reward in simulated dialogues. Error bars represent 95% confidence intervals.*

policy for one second in the "silence" state where the last speaker is the user and both parties remain silent. Note that this silent period could be either a short pause within the user's utterance or an end-of-utterance. The probability of executing the *speak* action increases rapidly about one second after entering the silence state. The probability of the system executing a *backchannel* action is more likely during the period 100ms to 500ms. This suggests that speakers usually make backchannels in preference to grabbing the floor during short pauses. In the given dialogues, acknowledgements such as "Okay" frequently occurred at the beginning of utterances, and thus the probability of a *backchannel* also increased after one second of silence.

Figure 4b shows the policy in "overlap" state where the system is speaking and the user barges in on the system. The probability of the *speak* action slowly decreases to avoid the overlap. However, it increases again after 800ms to model the cases where the system did not hand the floor back to the user. In Figure 4c, the policy in the state where the system is listening to the user is shown. The *speak* and *backchannel* actions have a tiny probability, and thus the system simply keeps listening with high probability.

Figure 5 shows how the policy shown in Figure 4a depends on the end-of-utterance (EOU) probability $\mathtt{e}$. Each line indicates the cumulative probability of *not* being silent by executing a *speak* or *backchannel* action. As expected, the learnt policy starts speaking or backchannels with higher probability as the EOU probability becomes higher. Interestingly, we also note that the cumulative probability before one second is larger when $\mathtt{e} < 0.1$ than when $0.3 \geq \mathtt{e} < 0.4$ (the blue solid line is higher than the red dotted line in Figure 5). This is because the conversant is more likely to generate backchannels than start speaking if the silence is believed to be a short pause within utterances.

Lastly, we compared the IRL policies with a simple handcrafted policy in terms of the resulting IRL rewards. The handcrafted policy takes the floor whenever the EOU probability is larger than a predefined threshold. Figure 6 compares rewards averaged over 30 simulated dialogues, using the IRL optimal deterministic policy (IRLdet), and the maximum-entropy randomised policy (IRLrand), and the handcrafted policy. Note that IRLdet outperformed IRLrand, as explained in Section 2.1. The handcrafted policy achieved the maximum average reward when the threshold is 0.9, but it was significantly worse than the IRL policies.

5. Conclusions

IRL is a technique for recovering an underlying reward function by observing demonstrated behaviours. The significance of IRL stems from its ability to learn from demonstrations in a diverse range of problems where an agent's behaviour can be characterised by a reward function which reflects the agent's objective and preferences. In this paper, we presented an application of IRL to micro-turn management, with the aim of identifying the conversational agent's objective function. Using annotated recordings of human-human dialogues, IRL successfully learnt the rewards from data and demonstrated better generalisation performance than supervised classification. In future work, the IRL reward function will be integrated into our POMDP-based dialogue system to allow live tests to be conducted with real users. We believe that IRL-based micro-turn management could be further improved by incorporating higher level features such as semantics and dialogue contexts [16] and this will also be studied in further work.

6. Acknowledgements

This work was supported by PARLANCE (www.parlance-project.eu), an EU Seventh Framework Programme project (grant number 287615).

7. References

[1] G. R. Jonsdottir, K. R. Thorisson, and E. Nivel, "Learning smooth, human-like turntaking in realtime dialogue," in *Proc. of 8th Int'l Conf. on Intelligent Virtual Agents (IVA)*, 2008.

[2] A. Raux, "Flexible turn-taking for spoken dialog systems," Ph.D. dissertation, Carnegie Mellon University, 2008.

[3] A. Raux and M. Eskenazi, "A finite-state turn-taking model for spoken dialog systems," in *Proc. of North American Chapter of the Association for Computational Linguistics - Human Language Technology (NAACL HLT)*, 2009, pp. 629–637.

[4] E. O. Selfridge and P. A. Heeman, "Importance-driven turn-bidding for spoken dialogue systems," in *Proc. of 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 177–185.

[5] D. Bohus and E. Horvitz, "Decisions about turns in multiparty conversation: From perception to action," in *Proc. of the 13th Int'l Conf. on Multimodal Interaction*, 2011.

[6] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, 2007.

[7] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. of 17th Int'l Conf. on Machine Learning (ICML)*, 2000, pp. 663–670.

[8] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, 1994.

[9] C. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proc. of 14th Int'l Conf. on Machine Learning (ICML)*, 1997.

[10] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. of 21th Int'l Conf. on Machine Learning (ICML)*, 2004.

[11] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. of 23rd AAAI Conf. on Artificial Intelligence*, 2008.

[12] S. Levine, Z. Popović, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Advances in Neural Information Processing Systems 24*, 2011.

[13] 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.

[14] SpaceBook. EC FP7/2011-16, grant number 270019. [Online]. Available: http://www.spacebook-project.eu

[15] S. Levine, Z. Popović, and V. Koltun, "Feature construction for inverse reinforcement learning," in *Advances in Neural Information Processing Systems 23*, 2010.

[16] A. Gravano and J. Hirschberg, "Turn-taking cues in task-oriented dialogue," *Computer Speech & Language*, vol. 25, no. 3, pp. 601–634, 2011.