On the use of social signal for reward shaping in reinforcement learning for dialogue management

Emmanuel Ferreira
LIA - University of Avignon
BP1228 - 84911 Avignon Cedex 9
Avignon, France
emmanuel.ferreira@univ-avignon.fr

Fabrice Lefèvre
LIA - University of Avignon
BP1228 - 84911 Avignon Cedex 9
Avignon, France
fabrice.lefevre@univ-avignon.fr

Abstract

This paper investigates the conditions under which social signals (facial expressions, postures, gazes, etc.), especially non-verbal multimodal user appraisal, can help to accelerate the learning capacity of a Reinforcement Learning (RL) agent in the dialogue management context. For this purpose a potential-based shaping reward method is used jointly with the Kalman Temporal Differences (KTD) framework so as to properly integrate the social aspects in an efficient optimization procedure through social-based additional reinforcement signals. Besides its general interest, this procedure could leverage system's development by allowing the designer to teach its system through explicit signals at its early stage of training. Experiments carried out using the state-of-the-art goal-oriented Hidden Information State (HIS) dialogue management framework in a simulation setup confirm the interest of the proposed approach.

.

1 Introduction

Goal-oriented statistical Spoken Dialogue Systems (SDSs), or even more generally Multimodal Dialogue Systems (MDSs), are the targets of this work. These systems are designed to achieve a task most often related to an information retrieval problem in collaboration with a human user (e.g. flight booking or hotel reservation services). The fundamental characteristic of this kind of "human-computer interface" is that the interaction between the human and the artificial agent (e.g. computer, robot, etc.) is mostly dominated by natural means of human communication (e.g. speech, gazes, gestures). The Dialogue Manager (DM) is the core component of SDSs, in charge of the interaction's course. It should infer the best decision sequence to fulfil the user goal. The dialogue management problem has first been described as a Markov Decision Process (MDP) in (Levin et al., 1997) and the Reinforcement Learning (RL) paradigm (Sutton and Barto, 1998) is employed to determine an optimal mapping between situations and actions, the policy. In this scheme the DM can be seen as an agent which has to interact with its environment (i.e. the user) in order to maximise some expected cumulative discounted reward. In most works the latter represents objective design criteria based on task completion and overall system efficiency. More recently, the MDP mathematical framework scheme was extended to Partially Observable Markov Decision Process (POMDP) to better cope with the inherent uncertainty on the information conveyed inside SDSs. This uncertainty comes from the fact that available pieces of information, collected from the user during consecutive dialogue turns, are extracted by error-prone input modules (e.g. speech recognizer, natural language understanding module, gesture recognizer, etc.). RL approaches were also successfully applied in this context (Young et al., 2010; Thomson and Young, 2010).

When developing a new SDS from scratch, in-domain dialogue corpora are seldom readily available and collecting such data is both time consuming and expensive (e.g. Wizard-of-Oz, prototyping). That is why, the capacity of a RL algorithm to learn online while interacting with the user is highly valuable. However, common approaches assume that an acceptable sub-optimal initial policy has been found by either exploiting user simulation methods (Schatzmann et al., 2005), or by hand (handcrafted dialogue manager) before any trials are made with real users. Recent works attempted to address this problem by using sample-efficient algorithms in order to limit

Proceedings of the 17th Workshop on the Semantics and Pragmatics of Dialogue, December 16-18, 2013, Amsterdam, The Netherlands.

the need of such a "bootstrap step". Thereby, TD-based SARSA with Gaussian Process (Gašić et al., 2010), incremental sparse Bayesian method (Lee and Eskenazi, 2012), or KTD (Daubigney et al., 2012) are among the most promising approaches. Anyhow, lowering the length of the warm-up learning phase, when the system can not interact with real users due to a high level of exploration and poor performance, is still an open problem when such systems are to be declined to real-world applications. One solution can be to introduce some initial expert knowledge (Williams, 2008) or to find ways to collect more hints from the environment which will accelerate the policy learning. For that purpose, we claim that social signals (Vinciarelli et al., 2009) can be employed as additional reinforcement signals (i.e. rewards) to refine and accelerate the policy optimization of a learning agent. Indeed, detecting social signals and social behaviours (e.g. emotions, turn taking attempts, politeness, noddings, postures, gazes, etc.) influence our everyday life behaviour in many ways (Custers and Aarts, 2005). Furthermore, by the fact that they can be gathered all along the dialogue, they may introduce a more granular view of the real quality of an interaction. Despite that some attempts to use emotion with RL have already been made (Broekens and Haazebroek, 2007), little has been done in the goal-oriented DM problem context. In this paper we propose a potential-based shaping reward method (Ng et al., 1999) to integrate these social aspects in combination with the use of the unified KTD framework with regards to its interesting properties (Geist and Pietquin, 2010; Daubigney et al., 2012). This preliminary study is carried out in a simulation setting where social reinforcement signals are simulated based on dialogue progress objective features representing the positiveness/negativeness of a particular situation. In this context, a better control over the experimental conditions, such as the simulated concept error rate level, is possible and comparison between several techniques is facilitated.

The remainder of the paper is organised as follows. In Section 2 some backgrounds on MDP/POMDP, RL paradigm, DM problem and KTD method are given. Then, in Section 3 social reward principle is detailed. Section 4 is dedicated to present the experimental setup. Then the following section details and comments on the different results obtained. Section 6 discusses on some considerations relevant to the use of social reinforcement, before concluding in Section 7 with some perspectives.

2 Background

This section briefly reviews the Markov Decision Processes (MDP) and the RL paradigm. Then, the casting of the DM problem as an MDP (POMDP) is presented. Finally, the KTD method is concisely introduced.

2.1 Markov Decision Processes

A tuple $\{S, A, T, R, \gamma\}$ forms a MDP, where S is the state space (discrete, continuous or mixed), A is the discrete action space, T is a set of Markovian transition probabilities, R is the immediate reward function, $R : S \times A \times S \rightarrow \Re$ and $\gamma \in [0, 1]$ the discount factor (discounting long term rewards). The environment evolves at each time step t to a state s_t and the agent picks an action a_t according to a policy mapping states to actions, $\pi : S \rightarrow A$. Then state changes to s_{t+1} according to the Markovian transition probability $s_{t+1} \sim T(.|s_t, a_t)$ and, following this, the agent received a reward $r_t = R(s_t, a_t, s_{t+1})$ from the environment. The overall problem of MDP is to derive an optimal policy maximising the reward expectation. Typically the averaged discounted sum over a potentially infinite horizon is used, $\sum_{t=0}^{\infty} \gamma^t r_t$. Thus, for a given policy and start state s, this quantity is called the value function: $V^\pi(s) = E[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, \pi] \in \Re^S$. V^* corresponds to the value function of any optimal policy π^*. The Q-function may be defined as an alternative to the value function. It adds a degree of freedom on the first selected action, $Q^\pi(s, a) = E[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi] \in \Re^{S\times A}$. As well as V^*, Q^* corresponds to the action-value function of any optimal policy π^*. If it is known, an optimal policy can be directly computed by being greedy according to Q^* , $\pi^*(s) = \arg\max_a Q^*(s, a) \forall s \in S$.

2.2 Dialogue Management as a POMDP

Dialogue management problem has first been described in (Levin et al., 1997) as a Markov Decision Process to determine an optimal mapping between situations and actions. The POMDP framework (Kaelbling et al., 1998), as a generalization of the fully-observable MDP, maintains a belief

distribution $b(s)$ over user states, assuming the true one is unobservable. Thereby, POMDP explicitly handles parts of the inherent uncertainty of the DM problem (e.g. word error rate, concept error rate). A POMDP policy maps the belief state space into the action space. That is why, the optimal policy can be understood as the solution of a continuous space MDP. In practice, POMDP problems are intractable to solve exactly due to the curse of dimensionality (i.e. belief state/action spaces). Among other techniques, the HIS model (Young et al., 2010) circumvents the RL scaling problem by organising the belief space into partitions, grouping states sharing the same probability, and then mapping the full belief space (partitions) into a much reduced summary space where RL algorithms work reasonably well.

Although variants have been proposed and tested, e.g. (Pinault and Lefèvre, 2011), HIS remains a reference. However, the choice of a Monte Carlo Control RL algorithm (Sutton and Barto, 1998) is still questioned and recent studies show the interest of considering sample-efficient algorithms for the DM problem (Gašić et al., 2010; Daubigney et al., 2012). More especially (Daubigney et al., 2012) showed that KTD framework offers a unified framework able to cope with all DM required properties: it is sample-efficient, it allows on-policy/off-policy learning through two algorithms (respectively KTD-Q and KTD-SARSA) which can both perform online and offline learning, it provides ways to deal with the "exploration/exploitation" dilemma using uncertainty on value estimate, it allows value tracking, and it supports linear and non-linear parametrisation. Furthermore, KTD algorithms were favourably compared to different state-of-the-art algorithms able to deal with one single property at once, such as Q-learning, LSPI or GP-SARSA.

2.3 The KTD Framework

The Kalman Temporal Differences (KTD) framework (Geist and Pietquin, 2010) is derived from the well-known Kalman filter algorithm (Kalman, 1960) aiming at inferring some hidden variables from related past observations and applied to the estimation of the temporal differences for the action-value function optimisation. In this framework, a parametric representation of the Q-function is chosen: $\hat{Q}_\theta = \theta^T \phi(s,a)$, where the feature vector $\phi(s,a)$ is a set of n basis functions to be designed by the practitioner and $\theta \in \Re^n$ the parameter vector to be learnt. Notice that just the very basic explanations are recalled here, for further details please refer to (Geist and Pietquin, 2010; Daubigney et al., 2012). The components of the parameter vector θ are the hidden variables which are modelled as a random vector. Such parameter vector is considered to evolve following a random walk though this evolution equation: $\theta_t = \theta_{t-1} + v_t$, with v_t a white noise of covariance matrix P_{v_t}. The latter allows to take into account the possible non-stationarity of the function. The observations correspond to the environment rewards which are linked to the hidden parameter vector through one of the sampled Bellman equations $g_t(\theta_t)$ depending on the RL scheme employed (i.e. evaluation for on-policy or optimality for off-policy learning):

$$g_t(\theta_t) = \begin{cases} \hat{Q}_{\theta_t}(s_t,a_t) - \gamma \hat{Q}_{\theta_t}(s_{t+1},a_{t+1}) \\ (evaluation) \\ \hat{Q}_{\theta_t}(s_t,a_t) - \gamma \max_a \hat{Q}_{\theta_t}(s_{t+1},a) \\ (optimality) \end{cases}$$

Rewards are supposed to follow the observation equation: $r_t = g_t(\theta_t) + n_t$ where a white noise n_t with covariance matrix P_{n_t} is also considered. Two algorithms can be defined: KTD-SARSA which denotes the use of the sampled evaluation Bellman equation and KTD-Q, the use of the sampled optimality one.

3 Social Reinforcement

In this section a rather simple definition of social reward is given followed by a mathematical formalisation of such a reward. Then, a method to simulate social signal is described.

3.1 Definition and formalisation

Social signal is a generic term which encompasses all the behavioural cues which can be encountered during an interaction with a human (e.g. blinks, smiles, crossed arms, laughter, nodding and the like). Social RL consists hence in exploiting these cues in order to guide the learning process. However, the agent can use this information in multiple ways: as reinforcement, as additional information integrated into the user state or as meta-parameter (e.g. in an exploration/exploitation scheme). Moreover, one may also think of using

emotion in the system response (*emotional agent*) and thus, make use of this information so as to improve its own social behaviour.

This work focuses on extracting social rewards based on positive and negative social signals emitted by the user and use them as additional rewards (or punishments). At each dialogue turn, a social reward may be perceived by the system. In this scenario a social signal can be seen as a user behaviour attesting its own judgement on the state evolution. Ergo, the social reward corresponds to the associated positiveness or negativeness of this signal represented as a signed real value. In that purpose we propose to consider the social reward function as a shaping reward function. The memoryless shaping reward function, which is one of the most general shaping pattern, is adopted here. So, the considered reward function is the sum of the basic environment reward function R_{env} (objective) and the new social one R_{social} (subjective). The resulting transformed MDP $M^{'}$ is defined by the tuple $(S, A, T, \gamma, R^{'})$ where $R^{'}$ is is the reward function defined as: $R^{'}(s_t, a_t, s_{t+1}) = R_{env}(s_t, a_t, s_{t+1}) + R_{social}(s_t, a_t, s_{t+1})$ where $R_{social} : S \times A \times S \rightarrow \Re$ is a bounded real-valued function called here the social-shaping reward function. Since the system is learning a policy for $M^{'}$ in the idea of using it in M, the question at hand is: what form of social-shaping reward function R_{social} can guarantee that the optimal policy in $M^{'}$ will be optimal in M? In the case where no further knowledge of T and R dynamics is available (no expert), a potential-based shaping reward leave (near-)optimal policies unchanged (Ng et al., 1999). Thereby, the potential-based shaping reward function is adopted for R_{social}, corresponding to function F in Ng et al.'s paper, and can be defined as follows:

$$R_{social}(s_t, a, s_{t+1}) = \gamma\psi(s_{t+1}) - \psi(s_t) \quad (1)$$

where ψ is a potential function, here computed using a heuristic score based on the social signal.

3.2 Social agenda-based simulation

3.2.1 Goal and agenda-based simulation

As indicated before, the social rewards are simulated in this study. We used the agenda-based user simulator presented in (Schatzmann et al., 2006) wherein the user is simulated at the intentional semantic level (i.e. dialogue act level). This approach factors the user state into an agenda A and

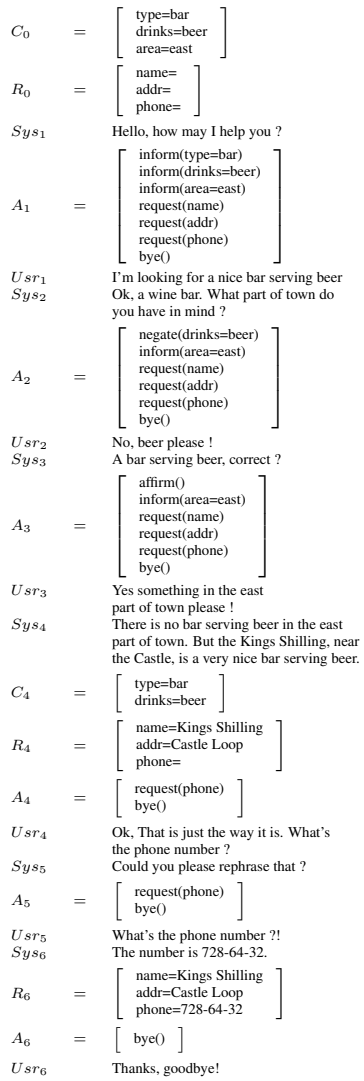

Figure 1: Sample dialogue and its corresponding agenda sequence

a goal G: $S = (A, G)$, where $G = (C, R)$. The goal G ensures that the simulated user reacts in an appropriate, consistent and goal-oriented manner. It consists of a set of constraints C specifying the required properties that the system should satisfy (they are the objects of the negotiation) and a set of requests R which represent the desired pieces of information (e.g. address, phone number, available schedules). The agenda A is a stack-like structure containing the pending user acts that are deemed necessary to elicit the information specified in the goal. For further details on this simulation method please refer to both (Schatzmann et al., 2006) and (Keizer et al., 2010).

Figure 1 illustrates how the simulation works and how tracking both the agenda and the goal evolution can serve to detect some situations when social signals are prone to be generated. For instance, in A_2, the presence of a *negate* act at the

top of the agenda means that a user constraint has been violated (here `drinks=beer`). So, it is a negative cue. In the same way, the `affirm` act in A_3 underlines a positive situation. That is why, the nature of the top dialogue act of the agenda can give an insight into the positiveness or the negativeness of the user state evolution.

3.2.2 Social cues

Table 1 presents some simple positive and negative cues extracted from the agenda and goal structures in the user simulator during dialogue simulations. Each of them is weighted in order to give more or less emphasis on specific features. Although a continuous scale is possible, a five-point agreement scale (Likert scale) is adopted here for ψ with regard to the way subjective measures are gathered in PARADISE (Walker et al., 1997). Each level is associated with a representative real number associated with an agreement scale, from strongly negative $(--)$ to strongly positive $(++)$. So, after a normalisation step the sum of all the simulated social features gives an overall score C_{s_t} which is rescaled on a five-point Likert scale using a threshold ξ. Thus, at each time step t, a "potential-like" social reward is computed using Eq 1 and ψ function:

$$\psi(s) = \begin{cases} -1 & ,if\ C_s < -\xi & (--) \\ -0.5 & ,if\ -\xi \leq C_s < 0 & (-) \\ 0 & ,if\ C_s = 0 & (neutral) \\ 0.5 & ,if\ 0 < C_s \leq \xi & (+) \\ 1 & ,if\ C_s > \xi & (++) \end{cases}$$

The process of social reinforcement reward computation can be decomposed into two steps. First, the gathering of positive and negative social cues from the factored user state. Second, the social reward estimation using the potential-based social reward function. An example of such a process is summarised in Table 2. The first column represents the analysed user state s_t (i.e. the corresponding agenda A_t and goal G_t in Fig. 1). The second and the third columns are respectively the lists of positive and negative cues which have been detected (using the id from Tab. 1) and their associated value in brackets. For example, in the first row and third column, cue 2 corresponds to the number of items in the agenda and the value 6 is extracted from A_3, minus sign indicates negativeness of the cue. The fourth column corresponds to the ψ value (i.e. the Likert score). It is computed applying some weights on the detected cue values. As an illustration, for the negative cue 3, $1/30$ is chosen as weight because the maximum number of turns allowed by the system is 30. Consequently, $1/30$ can be viewed as a normalisation value. It is important to notice that such weights have been determined following some expert intuitions. They have been chosen to correspond to an average user appraisal of the dialogue progress. In (Ferreira and Lefèvre, 2013), different user profiles are designed by varying these weights to study to what extent social signals can help user adaptation capacities of a learning agent. The last column shows the resulting social reward applying Eq. 1 with $\gamma = 0.95$. The positive score 0.45 denotes a quite favourable evolution between s_3 and s_4. To compete with the environment reward the social reward can be rescaled using an exponent. In real applications social cues could be elicited using several multimodal social detectors (e.g emotion face tracking, gesture classification, social keyword spotting). These latters may produce a list of detector-specific positive and negatives cues. For instance, the face tracker may produce a cue dedicating to smile detection which value is the probability of its inner model thereby consisting in a positive cue, likewise the definition of two lists of negative/positive keywords may help to produce two polarized cues from their detection in the ASR results associated with their posterior probabilities. Then, the same mecanism of a weighted interpolation could be used to infer $\psi(s)$ from the valued cues output by the various detectors.

Positive Cues	Negative Cues
1 Positive top dialogue act type (e.g. affirm, confirm)	1 Negative top dialogue act type (negate, deny, etc.)
2 Number of slots filled	2 Agenda size
3 Partial completion flag	3 Dialogue length
4 Final completion flag	4 Top agenda act contains already transmitted item

Table 1: List of positive and negative cues collected from agenda and goal

s_t	Positive cues	Negative cues	$\psi(s_t)$	R_{social}
s_3	1(1)	2(-6), 3(-4)	0.5	0.45
s_4	2(2/3) 3(1)	2(-2) 3(-5)	1	

Table 2: Social reward computation example

4 Experimental Setup

First, the HIS-based Dialogue System is briefly described. Then, some details on experimental conditions are given.

4.1 TownInfo Dialogue System

The TownInfo Dialogue System (Young et al., 2010) is a HIS-based dialogue system for the tourist information domain, related to a virtual town. The TowInfo system has already been tested with real users in (Schatzmann et al., 2006), and in a more recent and matured version, called CamInfo (Cambridge tourist information), in (Gašić et al., 2010). In order to deal with large state and action space the system maintains a set of partitions which represent the overall belief state. Both the latter and the action space are mapped into more reduced summary spaces where RL algorithms are tractable. The summary state space is the compound of two continuous values (the two-first top partitions probabilities) and three discrete values (last user act type and a partition and a history status). The summary action space contains 11 actions (e.g. inform, confirm). The environment rewards penalised each dialogue turn by -1 and at the end of a dialogue the DM is rewarded a +20 bonus if the goal is reached, nil otherwise.

4.2 Experimental details

To assess the performance of introducing social cues as a reinforcement signal, the online version of the off-policy KTD-Q algorithm (noted KTD-Q BASELINE) is employed as our baseline due to its high performance in the conditions at hand (Daubigney et al., 2012). The Q-function is parametrised using linear-based Radial Basis Function (RBF) networks, one per action, as described in (Daubigney et al., 2012) and the Bonus-Greedy scheme (Daubigney et al., 2011) is adopted, with $\beta = 1000$ and $\beta_0 = 100$. The discount factor γ is set to 0.95 in all experiments. By default, the user simulator is set to interact with the DM at a 10% concept error rate. The weight coefficient of the overall social reward is set to 4 and $\xi = 0.3$, likewise all other individual cues are weighted manually. All the results are averaged over 50 independent training under online RL conditions and are presented in terms of mean discounted cumulative rewards with respect to both the number of training dialogues (i.e. samples) or different CER levels. The associated standard deviations are added to all the results. The authors consider that the average cumulative environment rewards can be sufficient metric to compare the different approaches. This is explained by the fact that in the environment reward function the success (full user goal completion) is rewarded by a +20 bonus and failure and elapsed time (turn) respectively punished by a 0 and -1. For comparison purposes all the experiments with a social reward presented in our plots are given in terms of the environment reward, R_{env}, only.

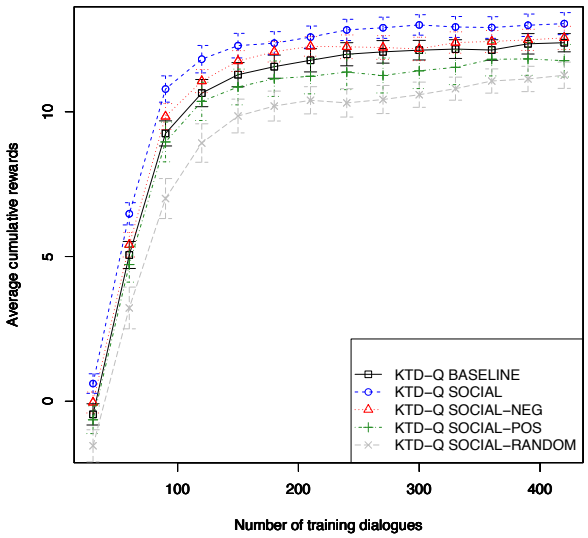

Figure 2: Results of 4 different configurations of the social-shaped KTD-Q algorithm compared to KTD-Q baseline during the learning of the policy (controlled case)

5 Results

This section presents the results obtained using the agenda-based user simulator described in Section 3.2.

5.1 Online policy using social reinforcement learning

In this section the benefits of adding social reinforcement signals for optimizing the DM policy are evaluated considering several social reinforcement configurations which take into account different kind of cues for the social reward computation. The classic approach noted KTD-Q SOCIAL considers both the negative and the positive social cues, as described in Section 3.2.2.

Results are shown in Figure 2 in terms of cumulative discounted environment rewards gathered during the learning stage of the policy (controlled case) when exploration is possible. For these curves, each point is an average of the 50 independent learning performance using a sliding window of 100 point width. Only the first 500 dialogues are considered here because we want to focus on the early stage of training for which system performance is critical. We can observe

that KTD-Q SOCIAL slightly outperforms KTD-Q BASELINE in terms of both the final learned performance, which is better of about 0.5 turn on average, and the learning time to achieve a similar performance level, which is reduced. For example, the performance obtained performing 200 dialogues with KTD-Q BASLINE algorithm are reached at about 100 dialogues using KTD-Q SOCIAL. Furthermore, a comparison between three other kinds of configuration of the simulated social signal is also made. The first (KTD-Q SOCIAL-NEG) and the second (KTD-Q SOCIAL-POS) configurations are respectively using only the negative or positive social cues. The third configuration is a randomized social signal generator (KTD-Q SOCIAL RANDOM). As expected, KTD-Q SOCIAL-RANDOM is the worst, followed by KTD-Q SOCIAL-POS, KTD-Q BASELINE and KTD-Q SOCIAL-NEG. KTD-Q SOCIAL which combines both positive and negative cues still obtains the best results. All configurations (except KTD-Q SOCIAL-RANDOM) are rather close if we consider the confidence radius of their results. However an important point is that even in the case of random social reinforcement, the potential-based technique ensures that convergence to the near-optimal policy is still preserved. From this experiment it seems that the convergence is better guided by negative information which is an interesting finding considering that negative emotions might be easier to emit and detect in a real setup.

5.2 Online policy in noisy conditions

Eventually we intend to evaluate the impact of noise on the proposed optimization procedure. Noise robustness is studied in terms of CER, Environment and Social Reward Error Rates, noted respectively ERER, SRER. Although the previous experiment has shown encouraging results when social reinforcement is considered, it should be kept in mind that in the previous conditions social signals are perfectly perceived by the learning agent. In a more realistic setup like user trials such signals, due to their inherent complexity (e.g. multimodal aspects, context-dependent interpretation) cannot be perfectly observed. This difficulty is introduced in the simulation by means of an artificial SRER. At a given rate the social cues are randomly modified to the inverse of what they should be. In the same way, when online learning is adopted the user should mark the overall dialogue in terms of

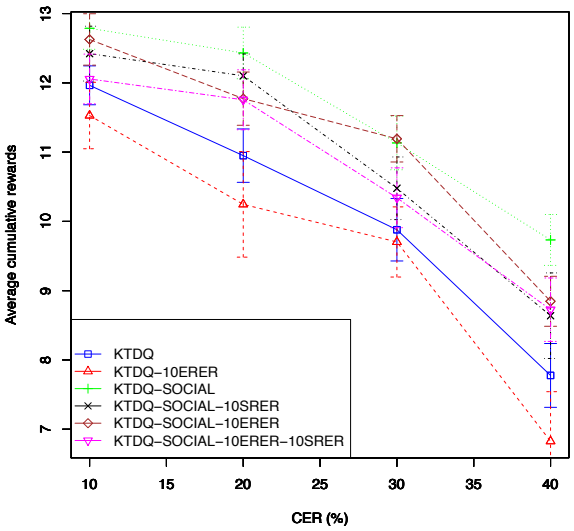

Figure 3: Results of baseline and social-shaped KTD-Q algorithms in different noise conditions (no control)

task completion (objective metric). But, as shown in (Gašić et al., 2010), the feedback given by a real user can be erroneous. This will be reflected by the ERER in our experiments. At a certain rate the final evaluation of dialogue success (correct or not) is inverted. Wrong feedbacks can be explained by the subjectivity of the task. Although the goal is achieved any inconsistent behaviour of the system during the dialogue can drive the user to penalise the system at the end, but also by the fact that a trial user is not really committed to the task, if the system fails there is no consequence for her or if the system asks for some constraint release the user has no personal rationale to guide her behaviour. In any case, the quality of the reward function is crucial for the RL algorithms as the speed of convergence to the optimal policy relies on it. In addition, the presence of high CER level also has a negative influence when this additional difficulty is present from the beginning of the learning (no progressive degradation).

Here, 7 methods are compared: KTD-Q BASELINE and KTD-Q BASELINE-10ERER, KTD-Q SOCIAL, KTD-Q SOCIAL-10ERER, KTD-Q SOCIAL-10SRER and KTD-Q SOCIAL-10ERER-10SRER. The 10XER mean that the corresponding error rate X is set to 10%. Results are shown in Figure 3 in terms of cumulative rewards with respect to different CER levels. For these curves, each point is an average made over the results obtained using 50 policies learned with 400 dialogues and then tested with 1000 dialogues. In the latter test setup, the next action is cho-

Use social	SRER	Rewards	Success rate
no	-	10.24 ($\pm$0.76)	91.14 ($\pm$1.58)
yes	0	11.77 ($\pm$0.38)	93.42 ($\pm$0.80)
yes	10	11.75 ($\pm$0.43)	93.73 ($\pm$0.58)
yes	20	11.28 ($\pm$0.45)	92.53 ($\pm$0.88)
yes	30	10.80 ($\pm$0.42)	91.68 ($\pm$1.10)
yes	40	10.67 ($\pm$0.43)	91.33 ($\pm$1.01)
yes	50	10.06 ($\pm$0.71)	89.34 ($\pm$3.70)

Table 3: Results of KTD-Q algorithm at 20% CER and 10 % ERER using different SRER levels (no control)

sen greedily with respect to the learnt Q-function (no exploration). Considering only the KTD-Q BASELINE and KTD-Q-BASELINE-10ERER the influence of CER and ERER can be easily identified. Thus, as the ERER and the CER increase the overall performance decreases. Nevertheless, in all conditions the use of a social reinforcement has a positive impact on the performance of the KTD-Q algorithm. Thus, social reinforcement improves the ability to defer the impact of noise in terms of both CER and ERER. One of the reasons for this is that social rewards are gathered all along the dialogue and offer a granular form of reward function. So, in case of the user giving an erroneous final reward, collected positive and negative social rewards can counterbalance this mistake (as an hint of the overall user satisfaction). Furthermore, in case of high CER, social rewards can favour or penalize a system local behaviour despite the overall task failure (or success). However, the benefit of social reinforcement tends to decrease as the SRER raises. Thereby, in order to study the impact of SRER alone, Table 3 is populated with the results obtained with different SRER levels at 20 % CER and 10 % ERER, both corresponding to realistic values for field trials. Above 30% SRER, taking into account social signals seems to be unnecessary or even disadvantageous. Actually, even if the results obtained with 40% SRER are slightly better than those obtained with the baseline, they do not converge as quickly (e.g. considering 200 training dialogues the baseline outperforms this social version). It is worth noting that ERER and SRER are simulated with no specific prior assumption. Indeed a rather simple random error approach is used. In more a sophisticated framework, such errors could be learnt from data.

6 Discussions

In this "proof of concept" study a simulation setup has been adopted, but undeniably real user trials are required to validate the suggested claims presented all along. Mechanisms to extract correctly social signals through multimodal cues from real user have to be envisaged as for instance what is done in the INTERSPEECH Computational Paralinguistics Challenge (Schuller et al., 2012). Even if the capacity of these methods remains highly imperfect if these cues are gathered in an unconstrained and implicit manner (Vinciarelli et al., 2009), the experiments in Section 5.2 show that we can evaluate them with a certain level of imprecision without jeopardizing the merits of the proposed method. Furthermore, we assume that this problem can be simplified if we consider an interaction with a cooperative and rational "seed user" (e.g. a system designer), which employs a limited set of non-verbal cues (e.g. head gesture, tone) in order to accelerate the learning process. The use of social rewards allows a more granular view of the reward function rather than a binary judgement at the end of the episode. So, it serves as a more specific way to avoid or strengthen some local system behaviours. Thereby, when sample-efficient algorithms are considered the approach can be viewed as a way to avoid the need for a user simulator by using 100-200 interactions with a seed user to bootstrap the system performance. Such setup can be assimilated to active learning like what is done in (Doshi and Roy, 2008) and thus linked to imitation-based (Price and Boutilier, 2003) or inverse approaches to RL as in (Chandramohan et al., 2011).

7 Conclusion

This paper has described a method by which social based reinforcement learning can be used to train a dialogue policy from scratch in just a few hundred dialogues and that improves the baseline performance in terms of rapidity of convergence. The approach also shows better robustness to noisy conditions in terms of semantic input error rate and environment reward error rate. The presented method also has interesting properties that guarantee the optimality when social signals are merged into an additional reinforcement learning signal using an amenable potential-based shaping reward function to introduce the detected social cues as additional reinforcement signals. In the present work the social signals were simulated from an agenda-based user simulator and thus real user trials are still needed to uphold our claims.

8 Acknowledgments

The authors would like to thank the Cambridge University Dialogue Systems Group for providing the TownInfo HIS System. As well as Lucie Daubigney, Olivier Pietquin and the MALIS Supélec-Metz Group for their help in using the KTD Framework. This work is partially funded by the ANR MaRDi project.

References

Joost Broekens and Pascal Haazebroek. 2007. Emotion and reinforcement: Affective facial expressions facilitate robot learning. In *Artificial Intelligence for Human Computing*, volume 4451 of *Lecture Notes in Computer Science*, pages 113–132.

Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, and Olivier Pietquin. 2011. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. In *Interspeech*.

Ruud Custers and Henk Aarts. 2005. Positive affect as implicit motivator: On the nonconscious operation of behavioral goals. *Journal of Personality and Social Psychology*, 89(2):129–142, August.

Lucie Daubigney, Milica Gašić, Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin, and Steve Young. 2011. Uncertainty management for on-line optimisation of a pomdp-based large-scale spoken dialogue system. In *Interspeech*.

Lucie Daubigney, Matthieu Geist, Senthilkumar Chandramohan, and Olivier Pietquin. 2012. A comprehensive reinforcement learning framework for dialogue management optimization. *Journal on Selected Topics in Signal Processing*, 6(8):891–902.

Finale Doshi and Nicholas Roy. 2008. Spoken language interaction with model uncertainty: an adaptive human–robot interaction system. *Connection Science*, 20:299–318.

Emmanuel Ferreira and Fabrice Lefèvre. 2013. Social signal and user adaptation in reinforcement learning-based dialogue management. In *IJCAI 2nd Workshop on Machine Learning for Interactive Systems*.

Milica Gašić, Filip. Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *SIGDIAL*.

Matthieu Geist and Olivier Pietquin. 2010. Kalman temporal differences. *Journal of Artificial Intelligence Research (JAIR)*, 39(1):483–532, September.

Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence Journal*, 101(1-2):99–134, May.

Rudolf E. Kalman. 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45.

Simon Keizer, Milica Gašić, Filip Jurčíček, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Parameter estimation for agenda-based user simulation. In *SIGDIAL*.

Sungjin Lee and M. Eskenazi. 2012. Incremental sparse bayesian method for online dialog strategy learning. *Journal on Selected Topics in Signal Processing*, 6:903–916.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *ASRU*.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*.

Florian Pinault and Fabrice Lefèvre. 2011. Unsupervised clustering of probability distributions of semantic graphs for pomdp based spoken dialogue systems with summary space. In *IJCAI 7th Workshop on knowledge and reasoning in practical dialogue systems*.

Bob Price and Craig Boutilier. 2003. A bayesian approach to imitation in reinforcement learning. In *IJCAI*.

Jost Schatzmann, Matt Stuttle, Karl Weilhammer, and Steve Young. 2005. Effects of the user model on simulation-based learning of dialogue strategies. In *ASRU*.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Engineering Review*, 21(2):97–126, June.

Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Müller, and Shrikanth Narayanan. 2012. Paralinguistics in speech and language - state-of-the-art and the challenge. *Computer Speech and Language (CSL), Special Issue on " Paralinguistics in Naturalistic Speech and Language"*, 27(1):4–39, Jan.

Richard S. Sutton and Andrew G. Barto. 1998. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588.

Alessandro Vinciarelli, Maja Pantic, and Hervé Bourlard. 2009. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. Paradise: a framework for evaluating spoken dialogue agents. In *ACL*.

Jason D. Williams. 2008. Integrating expert knowledge into pomdp optimization for spoken dialog systems. In *Proceedings of the AAAI-08 Workshop on Advancements in POMDP Solvers*.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.