Gestion de l'incertitude pour l'optimisation en ligne d'un gestionnaire de dialogues parlés à grande échelle basé sur les POMDP

Lucie Daubigney[1], Senthilkumar Chandramohan[1], Matthieu Geist[1], and Olivier Pietquin[1,2]

1. Supélec - Metz Campus, IMS Research Group, France
2. UMI 2958 (CNRS - GeorgiaTech), France

Résumé : L'utilisation de l'apprentissage par renforcement (AR) fait maintenant partie de l'état de l'art dans le domaine de l'optimisation de gestionnaires de dialogues parlés. Cependant avec cette méthode, entraîner un gestionnaire de dialogues requiert la génération de nombreuses données. C'est pourquoi beaucoup d'attention a été portée à la simulation d'utilisateurs ces dix dernières années. L'optimisation est donc faite avant de confronter le système à des utilisateurs réels et l'apprentissage est soit stoppé ou soit très lent durant l'utilisation pratique. Pendant ce temps-là, la recherche en AR a développé des algorithmes efficaces en termes d'échantillons. Par exemple les processus gaussiens pour l'AR ont récemment été appliqués aux gestionnaires de dialogues. Pour augmenter la vitesse d'apprentissage, l'incertitude sur les estimations calculées durant l'apprentissage est utilisée pour diriger l'exploration. Une comparaison entre différents schémas d'exploration montre que des améliorations significatives peuvent être apportées et qu'une optimisation en ligne rapide et sûre est possible, même sur une tâche complexe.

1 Introduction

Spoken Dialogues Systems (SDS), and especially task-oriented SDSs, are now very common and can be encountered in everyday life (hotline, call routing *etc.*). The difficulty when building such a dialogue system, apart from speech and language processing, is to build a dialogue manager that fulfil the user requests in the most efficient and natural way. To achieve optimality, dialogue strategies should thus handle speech and language processing errors in a natural way, introducing implicit and explicit confirmation as well as disambiguation subdialogues. Dialogue management can therefore be seen as a sequential decision making problem where decisions about which dialog act has to be performed in a given context (dialogue state) so as to maximize some optimality criterion. Sequential decision making is often addressed by Reinforcement Learning (RL) Sutton & Barto (1998) which is a statistical machine learning method that has been successfully applied to SDS during the last decade Singh *et al.* (1999); Levin *et al.* (2000); Pietquin & Dutoit (2006); Young *et al.* (2010). Optimizing dialogue management by means of RL requires casting the problem into a Markov Decision Process (MDP) Levin *et al.* (2000) or a Partially Observable MDP (POMDP) Young *et al.* (2010). In the later case, decisions are not taken according to a single inferred context but according to a distribution over different dialogue state hypotheses, which makes the learning process more robust to speech recognition and understanding errors.

Although the (PO)MDP framework is appealing for SDS optimisation, standard RL algorithms need large amounts of data to converge to an optimal strategy. Datasets expansion by means of user simulation has thus been intensively studied during the last 15 years Eckert *et al.* (1997); Pietquin & Dutoit (2006); Schatzmann *et al.* (2006). Yet, user simulation induces a modeling bias which is hard to predict Schatzmann *et al.* (2005) and can lead to suboptimal strategies after learning. In the meantime, RL research has made a lot of progress and especially batch algorithms that can learn from low amounts of fixed data have recently been shown to efficiently apply to SDS Pietquin *et al.* (2011). Anyway, examples of interactions used for learning may not represent all the types of users and learning should be pursued during the use of the system so as to take advantage of novel interactions. Recently, online learning with an efficient RL algorithm, namely GP-SARSA Engel *et al.* (2005), has been proposed to optimize dialogue management strategies Gašić *et al.*

(2010). GP-SARSA makes use of Gaussian processes Rasmussen & Williams (2006) to approximate the state-action value function from which is derived an optimal policy.

Several approaches can be envisioned to learn online an optimal policy depending on the way the policy is modified during the learning. If the initial policy is modified after each interaction with a user, the approach is called *on-policy*. Whereas when a policy is improved by using interactions collected with another policy, the approach is called *off-policy*. While off-policy learning offers the possibility to use a non-optimal but acceptable policy during learning, it is not obvious to identify the best moment when to switch to the learnt policy. On the other hand, on-policy learning (like in Gašić *et al.* (2010)) allows the policy to be enhanced immediately after each interaction but requires the user to undergo the changes made to the policy, even if they are not enhancements. The learning should therefore be made as quick as possible to avoid unacceptable changes to the policy to be seen by the users. This also poses the well-known exploration vs exploitation dilemma : should the system exploit its current knowledge and stop learning with the risk of being suboptimal or should it continue exploring different possible strategies to learn a better one with the risk of acting dangerously.

In this paper, we propose to use the uncertainty information on intermediate estimates (namely the Q-function) computed by the Gaussian process for RL method to build a sample-efficient online learning algorithm. Efficiency is reached thanks to different exploration schemes based on the computed uncertainty, allowing to test wisely unseen situations, not only because they are uncertain but also because they are expected not to be armful. The results are compared with Gašić *et al.* (2010). The rest of the paper is organized as follows. In Section 2, the problem of casting SDS optimization into the POMDP framework is presented. The following section proposes several exploration schemes based on uncertainty about estimates. In Section 4 some results are presented on a large-scale dialogue systems. Eventually, conclusions are drawn and perspectives proposed.

2 Dialogue Management as a POMDP

Dialog management (DM) can be seen as a sequential decision making problem. The decision maker is the dialog manager. From user acts (*observations*), it should choose and perform a system act (*actions*) in order to satisfy the user in an efficient and natural way. This satisfaction is quantified by a *reward* provided at the end of a dialogue and computed as a mixture of objective measures (such as the task completion, the dialogue duration *etc.*). Framed like this, DM can be cast as a POMDP (Partially Observable Markov Decision Process) : decision should be taken according to the full history of user and system acts. However, this history can briefly and efficiently be summarized by the hidden information state paradigm Young *et al.* (2010). Thus, DM can be cast as a continuous state MDP or a POMDP.

What is searched for is a policy π associating an action to each state, its quality being quantified by the so-called Q-function that gives the expected cumulative reward for starting with a given state-action pair and then following the policy π :

$$Q^{\pi}(s,a) = E[\sum_{i\geq 0} \gamma^i r_i | s_0 = s, a_0 = a, \pi],$$

γ being the discount factor, (s,a) the state-action pair and $(r_i)_{i\geq 0}$ the set of obtained rewards. The optimal policy (π^*) is greedy relatively to its Q-function ($Q^*(s,a)$) : $\pi^*(s) = \arg\max_a Q^*(s,a)$. The problem of finding the optimal policy thus resumes to the learning of the optimal Q-function. The Q-function allows comparing two policies, but also comparing two actions for a given state under a fixed policy. Usually, the state-action space is too large to allow an exact computation of the Q-function and approximation is mandatory.

Here we are interested in learning an optimal control policy while interacting with the user (online and on-policy learning). There are usually two steps : estimating the Q-function of the followed policy (with SARSA Sutton & Barto (1998) for example, or with GP-SARSA Engel *et al.* (2005) here) and choosing actions according to this estimated Q-function (the control part).

Let's assume that the Q-function approximation algorithm is provided. Choosing actions according to the current Q-function estimate is known as the dilemma between exploration and exploitation. At each interaction, the RL agent should choose between acting according to its current (imperfect) representation of the world (here the estimated Q-function) and performing some exploration action, suboptimal according to the current estimates but which can improve them. A classical but crude scheme is the use of an ϵ-greedy

policy : a greedy action resp. to the estimated Q-function ($\arg\max_a Q(s,a)$) is chosen w.p. ϵ, and a random one w.p. $1-\epsilon$. If learning from interactions with real users is envisioned, the exploration/exploitation dilemma is crucial. First, learning should be fast : the DM improves quickly and is able to adapt itself to users. Second, learning should be safe : the DM doesn't choose repetitively bad actions. This suggests that choosing actions purely randomly as done by the ϵ-greedy policy is not wise.

3 Gaussian Processes and Exploration

Here we pursue a work initiated by Gašić *et al.* (2010); Geist & Pietquin (2011). The underlying idea is that, if one has access to some uncertainty information about the estimated Q-function, than it should provide useful information for handling the dilemma between exploration and exploitation. For example, assume that two actions are possible for a given state, one (say a_1) having a higher estimated value than the other one (a_2). A greedy agent would choose a_1. But assume also that some confidence intervals about these estimates are also available. If uncertainty about a_2 is high, it should mean that this action is possibly better than a_1, and it can be worth trying it. On the other hand, if this confidence interval is small, one should choose a_1. We propose exploration schemes making such sort of choice automatically. Notice that we are interested in the uncertainty of the estimate, and not in the variance of the stochastic process of which the Q-function is the mean, which are two different things.

Two model-free value function approximators provide such an uncertainty information, namely KTD Geist & Pietquin (2010a) and GP-SARSA Engel *et al.* (2005), the later being considered in this paper. The underlying principle of GP-SARSA is to model the state-action value function as a Gaussian process (that is a set of jointly Gaussian random variables, these random variables being here the values of each state-action pair). A generative model linking rewards to values via the sampled Bellman evaluation operator and an additive noise is set. The Gaussian distribution of a state-action pair value conditioned on past observed rewards is computed by performing Bayesian inference and the value of this state-action pair is estimated as the mean of this Gaussian distribution. The associated variance quantifies the uncertainty. As any Bayesian method, a prior must be defined. For GP-SARSA, this is done through the choice of a kernel function defining the prior correlations of Q-function values.

We do not provide more details here, it should be enough to say that it provides at each step i an estimate $\hat{Q}_i(s,a)$ of the Q-function as well as an associated standard deviation $\hat{\sigma}_i(s,a)$. The ϵ-greedy policy does not use this uncertainty information : the greedy action $a_{i+1} = \arg\max_a \hat{Q}_i(s_{i+1},a)$ is chosen w.p. $1-\epsilon$ and a (uniform) random action otherwise. The obtained transition is then used to learn the new estimate $\hat{Q}_{i+1}$ (and $\hat{\sigma}_{i+1}$). Under some conditions, notably a decaying exploration factor, it converges to the optimal policy. However, it can be rather slow and unsafe during the exploration stage.

In Gašić *et al.* (2010) an *informed exploration* policy, also called *active learning* is introduced and is used here as a baseline. It chooses the greedy action (resp. to the estimated Q-function) $a_{i+1} = \arg\max_a \hat{Q}_i(s_{i+1},a)$ w.p. $1-\epsilon$ and another greedy action (resp. to the computed variance) $a_{i+1} = \arg\max_a \hat{\sigma}_i^2(s_{i+1},a)$ w.p. ϵ :

$$a_{i+1} = \begin{cases} \arg\max_a \hat{Q}_i(s_{i+1},a) \text{ w.p. } 1-\epsilon \\ \arg\max_a \hat{\sigma}_i^2(s_{i+1},a) \text{ w.p. } \epsilon \end{cases} \quad (1)$$

Therefore, the underlying idea is to act greedily respectively to the estimated state-action value function, and to choose sometimes an exploratory information. By choosing the less certain action, this scheme actually chooses the action which provides more information. Moreover, the exploration parameter ϵ is decayed to zero as learning progresses, so this scheme tends to the greedy policy. This improves over the classical ϵ-greedy policy but one possible drawback is that the exploration stage does not take the estimated value into account. Therefore, an action with a very low estimated value (which is possibly a "dangerous" action) can be chosen repeteadly, as long as it has a higher uncertainty that other actions (even if the difference is slight). We propose to use other exploration schemes.

The *confident-greedy* policy consists in acting greedily according to the upper bound of an estimated confidence interval Kaelbling (1993). For a tabular representation (for which the confidence interval width is proportional to $\frac{1}{\sqrt{n(s,a)}}$, $n(s,a)$ being the number of visits to the considered state-action pair), some PAC (Probably Approximately Correct) guarantees can be provided Strehl & Littman (2006). In the case of continuous state spaces (which occurs in SDS optimization), the state-action pairs being uncountable, this

approach does not hold. Here, standard deviation is provided by a Bayesian method, the prior is given and the posterior is computed. The distribution being Gaussian by assumption, the confidence interval width is proportional to the estimated standard deviation (which is actually true for any distribution, according to the Bienaymé-Tchebychev concentration inequality). Let α be a free positive parameter, we define the confident-greedy policy as :

$$a_{i+1} = \arg\max_a(\hat{Q}_i(s_{i+1}, a) + \alpha\hat{\sigma}_i(s_{i+1}, a)) \tag{2}$$

This strategy favors less certain actions if they correspond to the upper bound of the confidence interval. On the other hand, if for a given state all standard deviations are equal, than the agent will act greedily respectively to $\hat{Q}_i$. Notice also that the variance information provided by GP-SARSA tends to decrease as rewards are well predicted and as actions are more and more experimented.

The second approach we consider is the *bonus greedy* policy, inspired from Kolter & Ng (2009). Using a Bayesian argument, they act greedily resp. to the estimated Q-function plus a bonus, this bonus being proportional to the inverse of the number of visits to the state-action pair of interest. As $\frac{1}{\sqrt{n(s,a)}}$ is proportional to the standard deviation in a frequentist tabular approach, we interpret $\frac{1}{n(s,a)}$ as a variance. The proposed bonus-greedy policy therefore uses a variance-based bonus and is defined as (β_0 and β being two free parameters) :

$$a_{i+1} = \arg\max_a(\hat{Q}_i(s_{i+1}, a) + \beta\frac{\hat{\sigma}_i^2(s_{i+1}, a)}{\beta_0 + \hat{\sigma}_i^2(s_{i+1}, a)}) \tag{3}$$

This strategy also favors less certain actions and tends to be greedy respectively to the estimated state-action value function if variances are close to each other. A similar strategy has been shown efficient in model-based reinforcement learning using Gaussian Processes Deisenroth *et al.* (2009).

4 Experiments

Experiments have been led with the Hidden Information State (HIS) dialogue manager (Young *et al.* (2010)). The task consists in a tourist information system, assisting user to find a venue in Cambridge which can have up to 12 attributes. The algorithm used for the dialogue management optimisation is GP-SARSA with a polynomial kernel for the summary state space and a Dirac kernel for the action space ; it is the same set-up for the Q-function approximator as in Gašić *et al.* (2010). To obtain enough dialogues, the dialogue manager interacts at the intention level with a simulated user and not with real ones. The experiments are thus reproducible and the diversity of the users is ensured. An artificial speech understanding error rate of 10% is added. A positive reward ($+20$) is given at the end of the dialogue if the DM managed to fulfil the user request and a penalty (-1), aiming at encouraging short dialogues, is given at each dialogue turn.

The efficiency of the different exploration/exploitation schemes has been compared with the one proposed in Gašić *et al.* (2010) called *active learning* and which we referred to as the *informed exploration* scheme in Section 3.

First, we compare the different schemes by leading the same experiments as in Gašić *et al.* (2010). In Gašić *et al.* (2010), the results are obtained by stopping the learning and performing trials by using the greedy policy (no exploration anymore). The three schemes are compared Figure 1. This provides a baseline but one has to remember that the performance of the greedy policy can be good although the online performance during learning can be quite bad. Indeed, a totally random policy combined to an off-policy algorithm would lead to a good greedy policy and the poorer results during the learning. This is not acceptable when interacting with real users.

That is why we compare these results with ones obtained in an *online* way to get rid of this drawback. We show the performances of the policy while learning, this policy using the exploration scheme. In other words, it is an online evaluation of the policy, happening during the interactions with the user that serve to the learning. This difference is important since the greedy policy can be very good although very bad actions have been tested during the learning. Yet, if one wants to learn online, very bad actions should be avoided. Also, the advantage of learning online is that there is no need to chose the moment when to switch to the greedy policy, it is implicit from the exploration/exploitation schemes proposed in Section 3 that the learnt policy is asymptotically greedy. One other advantage of working online is that the policy keeps improving and adapting all along the interaction with the users. The different schemes are compared Figure 2. The

FIG. 1 – Performances of greedy policy. To get the average reward, each 100 dialogues, 100 policies were learnt and tested in a greedy way 1000 times each.

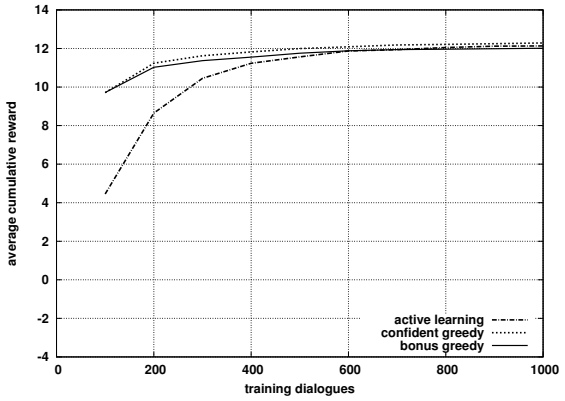
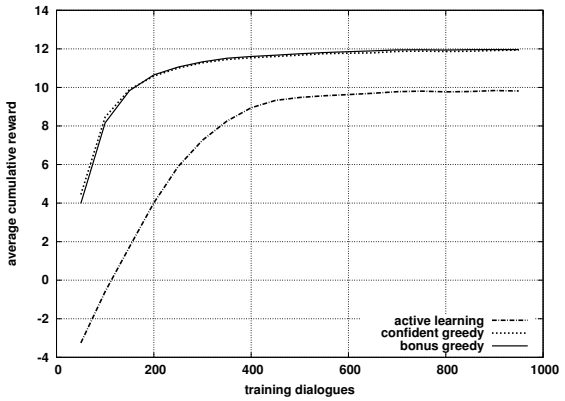


FIG. 2 – Performances of the policies. Parameters : $\alpha = 12$, $\beta_0 = 100$, $\beta = 1000$, $\epsilon_0 = 0.1$. The curves are obtained for an average over 1000 training trials. An average is made with a sliding window of 100 points width each 50 points.



average reward obtained by using the currently learnt policy has been plotted with respect to the number of dialogues already experienced during the learning. The parameters α, β and β_0, used in the two new exploration schemes, are chosen empiricallay to have good results but they are quite easy to tune since their order of magnitude can be deduced from the variance approximation of the Q function. The parameter ϵ_0 is used at the beginning of the learning in the active learning scheme and then decrease during the learning phase according to the number of dialogues experienced.

The comparison of the curves shows that improvements are brought when the information of uncertainty is used considering also the estimated value of the chosen action as proposed by the two novel exploration schemes. First, the number of samples needed to reach the same quality of policy is smaller (about 150 dialogues with respect to 600). This is an advantage when possibly interacting with real users, since the quicker a good policy is learnt the less the user is annoyed by an unadapted behaviour of the dialogue manager. Consequently, the user simulation needed to build data to learn a correct policy before interacting with real users may not be mandatory anymore. That removes the bias inherent with the simulation. Secondly, the asymptote for a big number of samples is higher (around 12 v.s around 10) with the new schemes proposed in Section 3. With *active learning*, the average number of steps to do the task is $20 - 10 = 10$ whereas with the new schemes it is $20 - 12 = 8$. The policy found seems better since the dialogue manager fulfils the user request with less steps, meaning that it better handles errors. This is due to the fact that learning online using the *active learning* scheme, the DM keeps performing uncertain actions which can lead to locally bad strategies. The two other schemes take the estimated value of the Q-function into account and avoid exploring uncertain actions if they are not expected to provide a correct behaviour. This is why the online

performances of the proposed schemes are better while the *active learning* scheme provides similar results when tested on the greedy policy. Anyway, one can notice that the online performances of the proposed exploration schemes are as good as the greedy one, while it is not the case for the *active learning* scheme.

5 Conclusion and Perspectives

In this paper, we have proposed two novel exploration schemes to learn (*online* and *on-policy*) an optimal dialogue management strategy by means of reinforcement learning. Based on this work, it is possible to envision learning of dialogue policies on real users, during the actual life of the system. This way, dialogue policies can permanently improve and are not subject to user modeling bias. It has been shown that the learning is made faster by the use of these exploration schemes which allows reaching an optimal policy after only a few hundreds of dialogues. These results outperform the results previously reported in Gašić *et al.* (2010).

In the future, we want to apply this method, as well as other RL algorithms providing uncertainty information which have been shown to be very efficient Geist & Pietquin (2010a,b) to direclty optimise the policy in interaction with real users. Moreover, these methods allow substantially more elements of the dialogue manager to be learnt which has the potential to make dialogue manager more data driven, flexible and human-like.

Références

DEISENROTH M., RASMUSSEN C. & PETERS J. (2009). Gaussian Process Dynamic Programming. *Neurocomput.*, **72**(7-9), 1508–1524.

ECKERT W., LEVIN E. & PIERACCINI R. (1997). User modeling for spoken dialogue system evaluation. In *Proc. ASRU'97*.

ENGEL Y., MANNOR S. & MEIR R. (2005). Reinforcement Learning with Gaussian Processes. In *Proceedings of the International Conference on Machine Learning (ICML 05)*.

GAŠIĆ M., JURČÍČEK F., KEIZER S., MAIRESSE F., THOMSON B., YU K. & YOUNG S. (2010). Gaussian processes for fast policy optimisation of POMDP-based dialogue managers. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 201–204 : Association for Computational Linguistics.

GEIST M. & PIETQUIN O. (2010a). Kalman Temporal Differences. *Journal of Artificial Intelligence Research (JAIR)*, **39**, 483–532.

GEIST M. & PIETQUIN O. (2010b). Statistically Linearized Least-Squares Temporal Differences. In *Proceedings of the IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, Moscow (Russia) : IEEE. 8 pages.

GEIST M. & PIETQUIN O. (2011). Managing Uncertainty within the KTD Framework. In *Proceedings of the Workshop on Active Learning and Experimental Design (AL&E collocated with AISTAT 2010)*, Journal of Machine Learning Research Conference and Workshop Proceedings, Sardinia (Italy).

KAELBLING L. P. (1993). *Learning in embedded systems*. MIT Press.

KOLTER J. Z. & NG A. Y. (2009). Near-Bayesian Exploration in Polynomial Time. In *international conference on Machine learning (ICML 09)*, New York, NY, USA : ACM.

LEVIN E., PIERACCINI R. & ECKERT W. (2000). A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, **8**(1), 11–23.

PIETQUIN O. & DUTOIT T. (2006). A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech and Language Processing*, **14**(2), 589–599.

PIETQUIN O., GEIST M., CHANDRAMOHAN S. & FREZZA-BUET H. (2011). Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*. accepted for publication - 24 pages.

RASMUSSEN C. E. & WILLIAMS C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

SCHATZMANN J., STUTTLE M. N., WEILHAMMER K. & YOUNG S. (2005). Effects of the user model on simulation-based learning of dialogue strategies. In *Proceedings of ASRU'05*.

SCHATZMANN J., WEILHAMMER K., STUTTLE M. & YOUNG S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, **21**(2), 97–126.

SINGH S., KEARNS M., LITMAN D. & WALKER M. (1999). Reinforcement learning for spoken dialogue systems. In *Proc. NIPS'99*.

STREHL A. L. & LITTMAN M. L. (2006). An Analysis of Model-Based Interval Estimation for Markov Decision Processes. *Journal of Computer and System Sciences*.

SUTTON R. & BARTO A. (1998). *Reinforcement learning : An introduction*. The MIT press.

YOUNG S., GASIC M., KEIZER S., MAIRESSE F., SCHATZMANN J., THOMSON B. & YU K. (2010). The hidden information state model : A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, **24**(2), 150–174.