k-Nearest Neighbor Monte-Carlo Control Algorithm for POMDP-based Dialogue Systems

F. Lefèvre*, M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu and S. Young
Spoken Dialogue Systems Group
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK
{frfl2, mg436, fj228, sk561, farm2, brmt2, ky219, sjy}@eng.cam.ac.uk

Abstract

In real-world applications, modelling dialogue as a POMDP requires the use of a summary space for the dialogue state representation to ensure tractability. Sub-optimal estimation of the value function governing the selection of system responses can then be obtained using a grid-based approach on the belief space. In this work, the Monte-Carlo control technique is extended so as to reduce training over-fitting and to improve robustness to semantic noise in the user input. This technique uses a database of belief vector prototypes to choose the optimal system action. A locally weighted k-nearest neighbor scheme is introduced to smooth the decision process by interpolating the value function, resulting in higher user simulation performance.

1 Introduction

In the last decade dialogue modelling as a Partially Observable Markov Decision Process (POMDP) has been proposed as a convenient way to improve spoken dialogue systems (SDS) trainability, naturalness and robustness to input errors (Young et al., 2009). The POMDP framework models dialogue flow as a sequence of unobserved dialogue states following stochastic moves, and provides a principled way to model uncertainty.

However, to deal with uncertainty, POMDPs maintain distributions over all possible states. But then training an optimal policy is an NP hard problem and thus not tractable for any non-trivial application. In recent works this issue is addressed by mapping the dialog state representation space (the *master* space) into a smaller *summary* space (Williams and Young, 2007). Even though optimal policies remain out of reach, sub-optimal solutions can be found by means of grid-based algorithms.

Within the Hidden Information State (HIS) framework (Young et al., 2009), policies are represented by a set of grid points in the summary belief space. Beliefs in master space are first mapped into summary space and then mapped into a summary action via the dialogue policy. The resulting summary action is then mapped back into master space and output to the user.

Methods which support interpolation between points are generally required to scale well to large state spaces (Pineau et al., 2003). In the current version of the HIS framework, the policy chooses the system action by associating each new belief point with the single, closest, grid point. In the present work, a k-nearest neighbour extension is evaluated in which the policy decision is based on a locally weighted regression over a subset of representative grid points. This method thus lies between a strictly grid-based and a point-based value iteration approach as it interpolates the value function around the queried belief point. It thus reduces the policy's dependency on the belief grid point selection and increases robustness to input noise.

The next section gives an overview of the CUED HIS POMDP dialogue system which we extended for our experiments. In Section 3, the grid-based approach to policy optimisation is introduced followed by a presentation of the k-nn Monte-Carlo policy optimization in Section 4, along with an evaluation on a simulated user.

*Fabrice Lefèvre is currently on leave from the University of Avignon, France.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 272–275,
Queen Mary University of London, September 2009.

2 The CUED Spoken Dialogue System

2.1 System Architecture

The CUED HIS-based dialogue system pipelines five modules: the ATK speech recogniser, an SVM-based semantic tuple classifier, a POMDP dialogue manager, a natural language generator, and an HMM-based speech synthesiser. During an interaction with the system, the user's speech is first decoded by the recogniser and an N-best list of hypotheses is sent to the semantic classifier. In turn the semantic classifier outputs an N-best list of user dialogue acts. A dialogue act is a semantic representation of the user action headed by the user intention (such as `inform`, `request`, etc) followed by a list of items (slot-value pairs such as `type=hotel, area=east` etc). The N-best list of dialogue acts is used by the dialogue manager to update the dialogue state. Based on the state hypotheses and the policy, a machine action is determined, again in the form of a dialogue act. The natural language generator translates the machine action into a sentence, finally converted into speech by the HMM synthesiser. The dialogue system is currently developed for a tourist information domain (Towninfo). It is worth noting that the dialogue manager does not contain any domain-specific knowledge.

2.2 HIS Dialogue Manager

The unobserved dialogue state of the HIS dialogue manager consists of the user goal, the dialogue history and the user action. The user goal is represented by a partition which is a tree structure built according to the domain ontology. The nodes in the partition consist mainly of slots and values. When querying the venue database using the partition, a set of matching entities can be produced. The dialogue history consists of the grounding states of the nodes in the partition, generated using a finite state automaton and the previous user and system action. A hypothesis in the HIS approach is then a triple combining a partition, a user action and the respective set of grounding states. The distribution over all hypotheses is maintained throughout the dialogue (*belief state monitoring*). Considering the ontology size for any real-world problem, the so-defined state space is too large for any POMDP learning algorithm. Hence to obtain a tractable policy, the state/action space needs to be reduced to a smaller scale summary space. The set of possible machine dialogue acts is also reduced in summary space. This is mainly achieved by removing all act items and leaving only a reduced set of dialogue act types. When mapping back into master space, the necessary items (i.e. slot-value pairs) are inferred by inspecting the most likely dialogue state hypotheses.

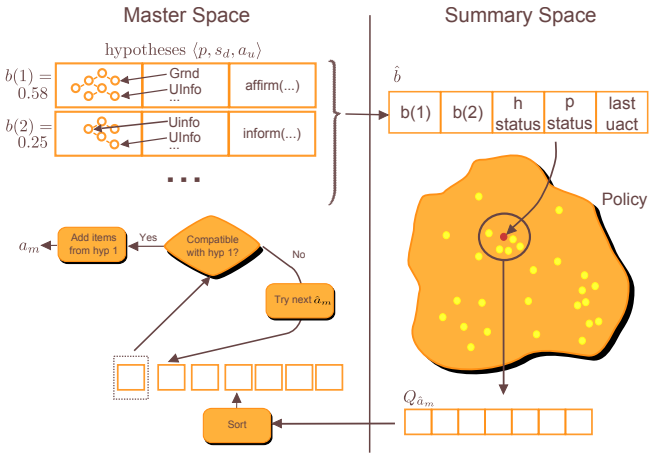

Figure 1: Master-summary Space Mapping.

The optimal policy is obtained using reinforcement learning in interaction with an agenda based simulated user (Schatzmann et al., 2007). At the end of each dialogue a reward is given to the system: +20 for a successful completion and -1 for each turn. A grid-based optimisation is used to obtain the optimal policy (see next section). At each turn the belief is mapped to a summary point from which a summary action can be determined. The summary action is then mapped back to a master action by adding the relevant information.

3 Grid-based Policy Optimisation

In a POMDP, the optimal exact value function can be found iteratively from the terminal state in a process called *value iteration*. At each iteration t, policy vectors are generated for all possible action/observation pairs and their corresponding values are computed in terms of the policy vectors at step $t-1$. However, exact optimisation is not tractable in practice, but approximate solutions can still provide useful policies. Representing a POMDP policy by a grid of representative belief points yields an MDP optimisation problem for which many tractable solutions exist, such as the Monte Carlo Control algorithm (Sutton and Barto, 1998) used here.

In the current HIS system, each summary belief point is a vector consisting of the probabilities of the top two hypotheses in master space, two discrete status variables summarising the state of the

Algorithm 1 Policy training with k-nn Monte Carlo

Let $Q(\hat{b}, \hat{a}_m)$ = expected reward on taking action $\hat{a}_m$ from belief point $\hat{b}$
Let $N(\hat{b}, \hat{a}_m)$ = number of times action $\hat{a}_m$ is taken from belief point $\hat{b}$
Let $\mathcal{B}$ be a set of grid-points in belief space, $\{\hat{b}\}$ any subset of it
Let $\pi_{\text{knn}} : \hat{b} \rightarrow \hat{a}_m; \forall \hat{b} \in \mathcal{B}$ be a policy
repeat
  $t \leftarrow 0$
  $\hat{a}_{m,0} \leftarrow$ initial greet action
  $b = b_0$ [= all states in single partition]
  Generate dialogue using ϵ-greedy policy
  repeat
    $t \leftarrow t + 1$
    Get user turn $a_{u,t}$ and update belief state b
    $\hat{b}_t \leftarrow \text{SummaryState}(b)$
    $\{\hat{b_k}\}_{\text{knn}} \leftarrow k\text{-Nearest}(\hat{b}_t, \mathcal{B})$
    $\hat{a}_{m,t} \leftarrow \begin{cases} \text{RandomAction} & \text{with probability } \epsilon \\ \pi_{\text{knn}}(\hat{b}_t) & \text{otherwise} \end{cases}$
    record $\langle \hat{b}_t, \{\hat{b_k}\}_{\text{knn}}, \hat{a}_{m,t} \rangle, T \leftarrow t$
  until dialogue terminates with reward R from user simulator
  Scan dialogue and update $\mathcal{B}$, Q and N
  for $t = T$ **downto** 1 **do**
    if $\exists \hat{b}_i \in \mathcal{B}$, $|\hat{b}_t - \hat{b}_i| < \delta$ **then** ← update nearest pt in $\mathcal{B}$
      for all $\hat{b}_k$ **in** $\{\hat{b}_k\}_{\text{knn}}$ **do**
        $w \leftarrow \Phi(\hat{b}_t, \hat{b}_k)$ ← Φ weighting function
        $Q(\hat{b}_k, \hat{a}_{m,t}) \leftarrow \frac{Q(\hat{b}_k, \hat{a}_{m,t}) * N(\hat{b}_k, \hat{a}_{m,t}) + R * w}{N(\hat{b}_k, \hat{a}_{m,t}) + w}$
        $N(\hat{b}_k, \hat{a}_{m,t}) \leftarrow N(\hat{b}_k, \hat{a}_{m,t}) + w$
      end for
    else ← create new grid point
      add $\hat{b}_t$ to $\mathcal{B}$
      $Q(\hat{b}_t, \hat{a}_{m,t}) \leftarrow R, N(\hat{b}_t, \hat{a}_{m,t}) \leftarrow 1$
    end if
    $R \leftarrow \gamma R$ ← discount the reward
  end for
until converged

top hypothesis and its associated partition, and the type of the last user act.

In order to use such a policy, a simple distance metric in belief space is used to find the closest grid point to a given arbitrary belief state:

$$|\hat{b}_i - \hat{b}_j| = \sum_{d=1}^{2} \alpha_d \cdot \sqrt{(\hat{b}_i(d) - \hat{b}_j(d))^2} + \sum_{d=3}^{5} \alpha_d \cdot (1 - \delta(\hat{b}_i(d), \hat{b}_j(d))) \quad (1)$$

where the α's are weights, d ranges over the 2 continuous and 3 discrete components of $\hat{b}$ and $\delta(x, y)$ is 1 iff $x = y$ and 0 otherwise.

Associated with each belief point is a function $Q(\hat{b}, \hat{a}_m)$ which records the expected reward of taking summary action $\hat{a}_m$ when in belief state $\hat{b}$. Q is estimated by repeatedly executing dialogues and recording the sequence of belief point-action pairs $\langle \hat{b}_t, \hat{a}_{m,t} \rangle$. At the end of each dialogue, each $Q(\hat{b}_t, \hat{a}_{m,t})$ estimate is updated with the actual discounted reward. Dialogues are conducted using the current policy π but to allow exploration of unvisited regions of the state-action space, a random action is selected with probability ϵ.

Once the Q values have been estimated, the policy is found by setting

$$\pi(\hat{b}) = \arg\max_{\hat{a}_m} Q(\hat{b}, \hat{a}_m), \qquad \forall \hat{b} \in \mathcal{B} \quad (2)$$

Belief points are generated on demand during the policy optimisation process. Starting from a single belief point, every time a belief point is encountered which is sufficiently far from any existing point in the policy grid, it is added to the grid as a new point. The inventory of grid points is thus growing over time until a predefined maximum number of stored belief vectors is reached.

The training schedule adopted in this work is comparable to the one presented in (Young et al., 2009). Training starts in a noise free environment using a small number of grid points and it continues until the performance of the policy asymptotes. The resulting policy is then taken as an initial policy for the next stage in which the noise level is increased, the set of grid points is expanded and the number of iterations is increased. In practice a total of 750 to 1000 grid points have been found to be sufficient and the total number of simulated dialogues needed for training is around 100,000.

4 k-nn Monte-Carlo Policy Optimization

In this work, we use the k nearest neighbor method to obtain a better estimate of the value function, represented by the belief points' Q values. The algorithm maintains a set of sample vectors $\hat{b}$ along with their Q value vector $Q(\hat{b}, a)$. When a new belief state $\hat{b}'$ is encountered, its Q values are obtained by looking up its k-nearest neighbours in the database, then averaging their Q-values.

To obtain good estimates for the value function interpolation, local weights are used based on the belief point distance. A Kullback-Leibler (KL) divergence (relative entropy) could be used as a distance function between the belief points. However, while the KL-divergence between two continuous distributions is well defined, this is not the case for sample sets. In accordance with the locally weighted learning theory (Atkeson et al., 1997), a simple weighting scheme based on a nearly Euclidean distance (eq. 1) is used to interpolate the policy over a set of points:

$$\pi_{\text{knn}}(\hat{b}) = \arg\max_{\hat{a}_m} \sum_{\{\hat{b}_k\}_{\text{knn}}} Q(\hat{b}_k, \hat{a}_m) \times \Phi(\hat{b}_k, \hat{b})$$

In our experiments, we set the weighting coefficients with the kernel function $\Phi(\hat{b}_1, \hat{b}_2) = e^{-|\hat{b}_1 - \hat{b}_2|^2}$.

Since it can be impossible to construct a full system act from the best summary act, a back-off strategy is used: an N-best list of summary acts, ranked by their Q values, is scrolled through until a feasible summary act is found. The resulting overall process of mapping between master and summary space and back is illustrated in Figure 1. The complete k-nn version policy optimisation algorithm is described in Algorithm 1.

The user simulator results for semantic error rates ranging from 0 to 50% with a 5% step are shown in Figure 2 for $k \in \{1, 3, 5, 7\}$, averaged over 3000 dialogues. The results demonstrate that the k-nn policies outperform the baseline 1-nn policy, especially on high noise levels. While our initial expectations are met, increasing k above 3 does not improve performances. This is likely to be due to the small size of the summary space as well as the use of discrete dimensions. However enlarging the summary space and the sample set is conceivable with k-nn time-efficient optimisations (as in (Lefèvre, 2003)).

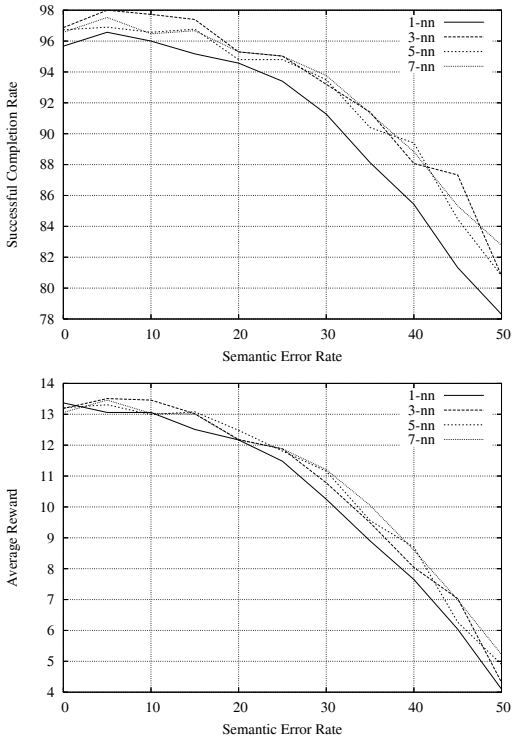

Figure 2: Comparison of the percentage of successful simulated dialogues and the average reward between the k-nn strategies on different error rates.

5 Conclusion

In this paper, an extension to a grid-based policy optimisation technique has been presented and evaluated within the CUED HIS-based dialogue system. The Monte-Carlo control policy optimisation algorithm is complemented with a k-nearest neighbour technique to ensure a better generalization of the trained policy along with an increased robustness to noise in the user input. Preliminary results from an evaluation with a simulated user confirm that the k-nn policies outperform the 1-nn baseline on high noise, both in terms of successful dialogue completion and accumulated reward.

Acknowledgements

This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and by the EU FP7 Programme under grant agreement 216594 (CLASSIC project: www.classic-project.org).

References

C Atkeson, A Moore, and S Schaal. 1997. Locally weighted learning. *AI Review*, 11:11–73, April.

F Lefèvre. 2003. Non-parametric probability estimation for HMM-based automatic speech recognition. *Computer Speech & Language*, 17(2-3):113 – 136.

J Pineau, G Gordon, and S Thrun. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc IJCAI*, pages pp1025–1032, Mexico.

J Schatzmann, B Thomson, K Weilhammer, H Ye, and SJ Young. 2007. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *HLT/NAACL*, Rochester, NY.

RS Sutton and AG Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Mass.

JD Williams and SJ Young. 2007. Scaling POMDPs for Spoken Dialog Management. *IEEE Audio, Speech and Language Processing*, 15(7):2116–2129.

SJ Young, M Gašić, S Keizer, F Mairesse, J Schatzmann, B Thomson, and K Yu. 2009. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, In Press, Uncorrected Proof.