Natural Belief-Critic: a reinforcement algorithm for parameter estimation in statistical spoken dialogue systems

F. Jurčíček, B. Thomson, S. Keizer, F. Mairesse, M. Gašić, K. Yu, and S. Young

Engineering Department, Cambridge University, CB2 1PZ, UK

{fj228, brmt2, sk561, f.mairesse, mg436, ky219, sjy}@eng.cam.ac.uk

Abstract

This paper presents a novel algorithm for learning parameters in statistical dialogue systems which are modelled as Partially Observable Markov Decision Processes (POMDPs). The three main components of a POMDP dialogue manager are a dialogue model representing dialogue state information; a policy which selects the system's responses based on the inferred state; and a reward function which specifies the desired behaviour of the system. Ideally both the model parameters and the policy would be designed to maximise the reward function. However, whilst there are many techniques available for learning the optimal policy, there are no good ways of learning the optimal model parameters that scale to real-world dialogue systems.

The Natural Belief-Critic (NBC) algorithm presented in this paper is a policy gradient method which offers a solution to this problem. Based on observed rewards, the algorithm estimates the natural gradient of the expected reward. The resulting gradient is then used to adapt the prior distribution of the dialogue model parameters. The algorithm is evaluated on a spoken dialogue system in the tourist information domain. The experiments show that model parameters estimated to maximise the reward function result in significantly improved performance compared to the baseline handcrafted parameters.

Index Terms: spoken dialogue systems, reinforcement learning, POMDP, dialogue management

1. Introduction

A POMDP dialogue manager includes three main parts: a dialogue model representing state information such as the user's goal, the user's dialogue act and the dialogue history; a policy which selects the system's responses based on the inferred dialogue state; and a reward function which specifies the desired behaviour of the system. In a POMDP system, the dialogue model provides a compact representation for the distribution of the unobserved dialogue state called the *belief state* and it is updated every turn based on the observed user inputs in a process called *belief monitoring*. Exact belief monitoring of the full dialogue state is intractable for all but the simplest systems. However, if the state is represented in the compact and approximate form of a dynamic Bayesian Network (BN), factored according to the slots in the system then by exploiting the conditional independence of the network nodes, a tractable system can be built [1]. In this case, the parameters of the model are the conditional distributions describing the nodes in the network.

The policy selects the dialogue system's responses (actions) based on the belief state at each turn, and it is typically trained using reinforcement learning with the objective of maximising the reward function. While there are many efficient techniques for learning the policy parameters [2, 3, 4], there are no good ways of learning the model parameters which scale to real-world dialogue systems. Hence, in virtually all current systems, the dialogue model parameters are handcrafted by a system designer [1, 3]. Ideally, one would like to estimate the parameters from the interactions with the user and some attempts have been made in this direction. For example, maximum likelihood estimates can be obtained by annotating the correct dialogue state in a corpus of real dialogues. However, in many real dialogues, some components of the dialogue state, especially the user's goal, are hard to determine. Hence, in practice this approach is restricted to cases where the user's goal remains constant and the dialogue is simple to annotate [5]. An alternative is to use algorithms such as Expectation-Maximization [6] or Expectation-Propagation [7] which can infer hidden state information. However, again these algorithms usually require the user goal to remain constant and even then it is not clear to what extent likelihood maximisation over a dialogue corpus correlates with the expected reward of the dialogue system.

This paper presents a novel reinforcement algorithm called Natural Belief-Critic (NBC) for learning the parameters of a dialogue model which maximise the reward function. The method is presented and evaluated in the context of the BUDS POMDP dialogue manager which uses a dynamic Bayesian Network to represent the dialogue state. However, the method is sufficiently general that it could be used to optimise virtually any parameterised dialogue model. Furthermore, unlike most of the maximum likelihood methods used so far, the NBC algorithm does not require that the user goal remains constant.

The paper is structured as follows. Section 2 briefly describes the BUDS dialogue manager and the method it uses for policy representation [1]. Section 3 then describes policy gradients methods and a specific form called the Natural Actor-Critic (NAC) algorithm which is used to optimise the BUDS policy. In Section 4, the proposed Natural Belief-Critic algorithm is presented as a generalisation of the NAC algorithm and then in Section 5 it is evaluated on a system designed for the tourist information domain. Finally, Section 6 presents conclusions.

2. BUDS dialogue manager

In a POMDP dialogue system, the true dialogue state s_t is unknown. Therefore, the policy selects an action a_t at time t based on the distribution over all states called the belief state, $b(s_t)$. The estimate of the belief state depends on past observations and actions. If the system is Markovian then the belief state b_t depends only on the previous belief state b_{t-1}, the current observation o_t and the last system action a_{t-1}:

$$b(s_t;\tau) = k{\cdot}p(o_t|s_t;\tau)\sum_{s_{t-1}} p(s_t|a_{t-1},s_{t-1};\tau)b(s_{t-1}|h_{t-1};\tau) \tag{1}$$

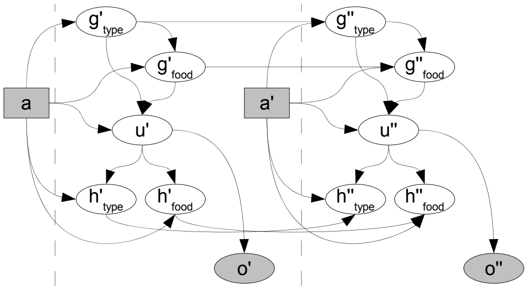

Figure 1: An example factorisation for the Bayesian network representing part of a tourist information dialogue system.

where the transition probability function $p(s_t|a_{t-1}, s_{t-1}; \tau)$ and the observation probability $p(o_t|s_t; \tau)$ represent the dialogue model which is parameterised by τ and k is a normalisation constant.

2.1. The dialogue model

A naive implementation of (1) is not tractable since there are billions of states in a real-world spoken dialogue system[1]. Thus, the BUDS dialogue manager uses a Bayesian Network (BN) to represent the state of the POMDP system, where the network is factored according to the slots in the system [1]. Provided that each slot or network node has only a few dependencies, tractable systems can be built and belief estimates maintained with acceptable accuracy using approximate inference [8].

The BUDS dialogue state is factored into three components: the user goal g, the user action u and the dialogue history h. In addition, the goal and the history are further factored into sub-goals g_i and sub-histories h_i according to a set of slots, $i \in \mathcal{I}$, in the system. For example, in a tourist information system typical sub-goals might be the type of venue required ("type") or the type of food ("food"). The sub-history nodes allow the system designer to store information about whether a user requested information or the system informed the user about some slot. The user action u is the estimate of the true dialogue act from the observation o.[2] Fig. 1 shows the resulting network for two time-slices of a two-slot system based on this idea.

The BN model parameters τ comprise the set of conditional probabilities of the node values. For example, the "food" sub-goal values are described by the probability $p(g''_{food}|g'_{food}, g''_{type}, a'; \tau_{food})$ parameterised by τ_{food}. To reduce the number of parameters specifying the distributions in the sub-goals, some parameters are tied together on the assumption that the probability of change in the sub-goals is constant given the last system action and the parent sub-goal. For example, the probability of change from "Chinese" to "Indian" in the sub-goal "food" is equal to the probability of change from "Chinese" to "Italian".

2.2. The Policy

The BUDS dialogue manager uses a stochastic policy $\pi(a|b; \theta)$ which gives the probability of taking action a given belief state b and policy parameters θ. When used in the dialogue manager, the policy distribution is sampled to yield the required action at each turn. To reduce complexity, for every action a, the belief state is mapped into a vector of features, $\Phi_a(b)$ and the policy is then approximated by a softmax function:

$$\pi(a_t|b(\cdot;\tau)); \theta) \approx \frac{e^{\theta^T \cdot \Phi_{a_t}(b(\cdot;\tau))}}{\sum_{\tilde{a}} e^{\theta^T \cdot \Phi_{\tilde{a}}(b(\cdot;\tau))}}. \tag{2}$$

To estimate the policy parameters, BUDS uses the Natural Actor-Critic (NAC) algorithm [4] (see Section 3).

A further reduction in complexity can be achieved by utilising summary actions [1]. For example, if the dialogue manager confirms the value of some sub-goal then it should always confirm the most likely value. As a result, the full set of actions is not needed. The mapping of the summary actions into full dialogue acts is performed by a handcrafted function based on the information in the belief state.

There are a variety of possible forms for the Φ function [2]. The BUDS dialogue manager uses factored grid-based approximation. In this case, for every node in the BN a set of binary features is generated based on the probabilities of two most likely values. BUDS also supports handcrafted policies which are designed by an expert. These policies deterministically choose which action to take given the features.

3. Policy gradients

The objective of reinforcement learning is to find a policy π which maximises the expected reward $J(\theta)$:

$$J(\theta) = E\Big[\frac{1}{T}\sum_{t=1}^{T} r(s_t, a_t) \mid \pi_\theta\Big],$$

where $r(s_t, a_t)$ is the reward when taking action a_t in state s_t.

Learning θ can be achieved by a gradient ascent which iteratively adds a multiple of the gradient to the parameters being estimated. Using "the log likelihood-ratio trick" and Monte Carlo sampling, the gradient can be estimated as follows:

$$\nabla J(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} \nabla \log \pi(a_t^n|b_t^n; \theta) R_n \tag{3}$$

where the sampled dialogues are numbered $n = 1, \ldots, N$, the n-th dialogue has a length of T_n turns, and $R_n = \frac{1}{T_n}\sum_{t=1}^{T_n} r(s_t, a_t)$ is the reward accumulated in dialogue n. To obtain a closed form solution for the gradient ∇J, the policy π must be differentiable w.r.t. θ. Conveniently, the softmax function in (2) is "linear" w.r.t. the parameters θ. Thus, it is easy to derive an analytic form for the gradient ∇J.

Although (3) can provide an estimate for the "vanilla" gradient, it has been shown that the natural gradient $\widetilde{\nabla} J(\theta) = F_\theta^{-1}\nabla J(\theta)$ is more effective for optimisation of statistical models where F_θ is the Fisher Information Matrix [9]. Based on this idea, Peters et al. developed the Natural Actor-Critic (NAC) algorithm which estimates a *natural gradient* of the expected reward function [4]. The appealing part of the NAC algorithm is that in practice the Fisher Information Matrix does not need to be explicitly computed. To obtain the natural gradient, w, of $J(\theta)$, NAC uses a least square method to solve the following set of equations:

$$R_n = \left[\sum_{t=1}^{T_n} \nabla \log \pi(a_t^n|b_t^n; \theta)^T\right] \cdot w + C \quad \forall n \in \{1, \ldots, N\}.$$

Once w has been found, the policy parameters can be iteratively improved by $\theta' \leftarrow \theta + \beta w$, where β is a step size.

Of all the policy optimisation algorithms tested with BUDS, the NAC algorithm has proved to be the most robust suggesting that the use of the natural gradient is critical. The question

[1] Note that if a dialogue system has 10 slots and each slot has 10 different values then there are 10^{10} distinct states.

[2] In the BUDS dialogue manager, the observations and system actions are implemented as dialogue acts. A dialogue act conveys the user or system intention (such as inform, request, etc) and a list of slot-value pairs (e.g. type=hotel, area=east).

therefore arises whether this type of policy gradient method can be generalised to optimise not just the policy but the parameters of the dialogue model as well.

4. Natural Belief-Critic algorithm

The difficulty with using policy gradient methods for learning the parameters of the dialogue model is that since the function Φ, which extracts features from the belief state, is usually a handcrafted function of non-continuous features, the policy is not usually differentiable w.r.t. τ. However, this problem can be alleviated by assuming that the model parameters τ come from a prior distribution $p(\tau;\alpha)$ that is differentiable w.r.t. the parameters α. This leads to a generalisation of the NAC algorithm called the Natural Belief-Critic (NBC) algorithm.

The goal of NBC is to learn the parameters α of the prior distribution while maximising the expected reward. The algorithm assumes that the policy is fixed during training. At each iteration, the NBC algorithm samples the model parameters, executes dialogues, and stores the rewards observed at the end of each dialogue. After collecting sufficient statistics, the algorithm updates the prior distribution based on the observed rewards. Finally, the expected values for τ given the distribution $p(\tau;\alpha)$ provide the new estimates for τ.

The techniques used in NAC to compute the natural gradient can be extended to the NBC algorithm since both algorithms sample from the distribution for which they are learning the parameters. The only difference is that NBC samples only at the beginning of a dialogue. As a result, NBC solves the following set of equations:

$$R_n = \nabla \log p(\tau^n;\alpha)^T \cdot w + C \quad \forall n \in \{1, \dots, N\} \qquad (4)$$

to obtain the natural gradient w of the expected reward.

In order to use NBC in practice a prior for the model parameters τ is needed. Since the parameters of the BN described in Section 2.1 are parameters of multiple multinomial distributions, a product of Dirichlet distributions provides a convenient prior.

Formally, for every node $j \in \{1, \dots, J\}$ in the BN, there are parameters τ_j describing a probability $p(j|par(j);\tau_j)$ where the function $par(j)$ defines the parents of the node j. Let $|par(j)|$ be the number of distinct combinations of values of the parents of j. Then, τ_j is composed of parameters of $|par(j)|$ multinomial distributions and it is structured as follows: $\tau_j = \left[\tau_{j,1}, \dots, \tau_{j,|par(j)|}\right]$. Consequently, a prior for τ_j can be formed from a product of Dirichlet distributions: $\prod_{k=1}^{|par(j)|} Dir(\tau_{j,k};\alpha_{j,k})$ parameterised by $\alpha_{j,k}$. Let the vector $\tau = [\tau_1, \dots, \tau_J]$ be a vector of all parameters in the BN. Then, the probability $p(\tau;\alpha)$ from (4) can be defined as $p(\tau;\alpha) = \prod_{j=1}^{J} \prod_{k=1}^{|par(j)|} Dir(\tau_{j,k};\alpha_{j,k})$ which has a closed form log-derivative w.r.t. α and can be used in (4) to compute the natural gradient w. The complete NBC algorithm is described in Algorithm 1.

5. Evaluation

An experimental evaluation of the Natural Belief-Critic algorithm was conducted using the BUDS dialogue system described in Section 2. The goal of the evaluation was to test whether the NBC algorithm could improve on a set of carefully handcrafted model parameters which had been refined over time to optimise performance. The evaluation was in two parts. Firstly a set of model parameters were estimated using a finely tuned handcrafted policy, and secondly, a set of model parameters were estimated using a stochastic policy trained using the NAC algorithm. In both cases, the results were compared to the performance obtained using the initial handcrafted model parameters.

Algorithm 1 Natural Belief-Critic

```
Let $\tau$ be the parameters of the dialogue model
Let $p(\tau;\alpha)$ be a prior for $\tau$ parameterised by $\alpha$
Let $\alpha_1$ be the initial parameters of the prior for $\tau$
Let $\pi$ be a fixed policy
Let $N$ be the number of dialogues sampled in each iteration
Let $M$ be the number of training iterations
Let $\beta$ be a step size

for $i = 1$ to $M$ do
    Collecting statistics:
    for $n = 1$ to $N$ do
        Draw parameters $\tau^n \sim p(\tau^n;\alpha_i)$
        Execute the dialogue according the policy $\pi$
        Observe the reward $R_n$
    end for
    Critic evaluation:
    Choose $w_i$ to minimize the sum of the squares of the errors of
        $R_n = \nabla \log p(\tau^n;\alpha)^T \cdot w_i + C$
    Parameter update:
    $\alpha_{i+1} \leftarrow \alpha_i + \beta w_i$
end for
```

The systems were trained and tested using an agenda based user simulator, for the Town-Info domain which provides tourist information for an imaginary town [1, 3]. The user simulator incorporates a semantic concept confusion model, which enables the systems to be trained and tested across a range of semantic error rates. The reward function used in all experiments awards 100 minus the number of dialogue turns for a successful dialogue and 0 minus the number of turns for an unsuccessful one.

5.1. Dialogue model for the Town-Info domain

The Bayesian Network for the Town-Info domain contains nine sub-goals: name of the venue, type of venue, area, price range, nearness to a particular location, type of drinks, food type, number of stars and type of music. Every sub-goal has a corresponding sub-history node. The network also has nodes to represent address, telephone number, a comment on the venue and the price. However, for these only their sub-history nodes are used since a user can only ask for values of these slots and cannot specify them as query constraints. Finally, the network has two special nodes. The "method" node stores the probability that the user is searching for a venue by constraint rather than by name. The "discourse" node infers whether a user wants the system to repeat the last system action, restart the dialogue, end the dialogue or provide the user with more information about the last offered venue. Although the dialogue manager does not ask about these nodes explicitly, their values are inferred just like any other node.

The history, "method", and "discourse" nodes use fully parameterised conditional probabilities in order to capture the detailed characteristics of dialogue flow. All of the other sub-goal nodes use parameter tying as described in Section 2.1. Overall this results in a total of 577 parameters in the dialogue model.

5.2. Experiments

Dialogue model parameters using the handcrafted policy were estimated by running the NBC algorithm for 50 iterations with the simulator set to give a 40% error rate. In each iteration, 16k dialogues were sampled. Both the baseline system and the system with the learnt BN parameters were evaluated over error rates ranging from 0% to 50%. At each error rate, 5000 dia-

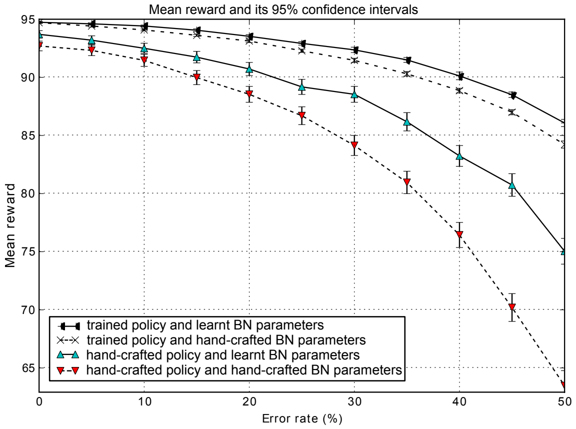

Figure 2: Comparison of the mean rewards of the handcrafted BN model parameters and the parameters learnt by NBC when trained using both a handcrafted policy and a trained policy.

logues were simulated and to reduce the variance of results, this training and evaluation procedure was executed 5 times. The averaged results along with 95% confidence intervals are depicted in Fig. 2. As can be seen, the system with trained BN parameters significantly outperforms the system with handcrafted parameters especially at high error rates. For example, at 40% error rate, the mean reward was increased by 8.3% ($p < 0.05$). Inspection of the results suggests that this improvement can be mostly attributed to the sub-optimality of the handcrafted policy and the ability of the learnt BN parameters compensate for this.

In the second experiment, the NBC algorithm was used to estimate a set of model parameters for a system using an optimised stochastic policy. The full training procedure was executed in three steps. First, a stochastic policy was learnt using a dialogue model initialised with handcrafted parameters. To train the policy, the NAC algorithm was executed for 200 iterations at a 40% error rate and in each iteration 4000 dialogues were simulated. Second, NBC was used to train BN parameters using the newly trained stochastic policy. Thirdly, the policy was retrained using NAC to take advantage of the improved model parameters. The final system was evaluated as in the first task; although in this case, the training and evaluation procedure was executed 20 times. The results, depicted in Fig. 2, show that a system with model parameters trained using NBC significantly improves on the system with handcrafted model parameters even when used with a trained policy. At 40% error rate, the mean reward was increased by 2.2% ($p < 0.05$). Further iterations of model parameter estimation and policy optimization did not lead to any further improvement in performance.

Inspection of the learnt model parameters compared to the handcrafted parameters based on KL-divergence showed that greatest effect of the NBC-based optimisation was on the "method" and "discourse" nodes. This is in line with expectations since the probabilities of change in these nodes are less intuitive and they are therefore much harder to set manually.

The NBC algorithm was also tested with initial model parameters different to the handcrafted ones. Simulations showed that although the algorithm is able to improve on arbitrary initialisations, the maximum performance achieved is sensitive to the initialisation, presumably because the algorithm converges to differing local optima.

Experiments were also conducted with uninformative (uniform) priors on the model parameters; though, they were not entirely successful since in this case, the final rewards were lower by 10%-20% in comparison with the rewards obtained when using the handcrafted parameters. It appears that the NBC algorithm too quickly reduces the variance of the prior distribution. Consequently, it limits exploration of the dialogue model parameters.

The NBC algorithm can also be understood as a random search algorithm. Thus, other state-of-the-art random search techniques such as SPSA [10] and CMA-ES [11] can be used. However, informal testing with these techniques yielded no further improvement to the results reported here.

6. Conclusion

This paper has proposed a novel method called the Natural Belief Critic algorithm for estimating the model parameters of a POMDP-based dialogue system so as to maximise the reward. Based on observed rewards obtained in a set of training dialogues, the algorithm estimates the natural gradient of the expected reward of a dialogue system and then adapts the Dirichlet prior distributions of the model parameters. Simulations have shown that the NBC algorithm significantly improves upon an initial set of handcrafted model parameters when used with both handcrafted and trained policies. Although the NBC algorithm converges reliably, the achievable maximum reward is sensitive to the initialisation. Thus the algorithm is most effective for improving on an existing set of model parameters which have either been handcrafted or estimated by other methods such as maximum likelihood.

Acknowledgment: 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).

7. References

[1] 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.

[2] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 1998.

[3] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management," *Computer Speech and Language, In press*, 2009.

[4] J. Peters, S. Vijayakumar, and S. Schaal, "Natural Actor-Critic," in *European Conference on Machine Learning (ECML)*. Springer, 2005, pp. 280–291.

[5] D. Kim, H. S. Sim, K. Kim, J. H. Kim, H. Kim, and J. W. Sung, "Effects of User Modeling on POMDP-based Dialogue Systems," in *Proceedings of Interspeech*, 2008.

[6] U. Syed and J. D. Williams, "Using automatically transcribed dialogs to learn user models in a spoken dialog system," in *HLT*, Morristown, USA, 2008, pp. 121–124.

[7] B. Thomson, "Statistical methods for spoken dialogue management," Ph.D. dissertation, University of Cambridge, 2010.

[8] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[10] J. C. Spall, *Introduction to Stochastic Search and Optimization*. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[11] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.