Active Learning in Partially Observable Markov Decision Processes

Robin JAULMES
McGill University
robin.jaulmes@mail.mcgill.ca

Joelle PINEAU
McGill University
jpineau@cs.mcgill.ca

Doina PRECUP
McGill University
dprecup@cs.mcgill.ca

Learning in Partially Observable Markov Decision Processes is a notoriously difficult problem. The goal of our research is to address this problem for environments in which a partial model may be available, in the beginning, but in which there is uncertainty about the model parameters. We developed and algorithm called MEDUSA [4,5], which is based on ideas from active learning [1,2,3].

We assume that prior knowledge about the model, as well as the level of uncertainty in the model, can be represented by a Dirichlet distribution over possible models. The parameters of this distribution are then updated whenever new experience is acquired. This allows a simple framework for combining a priori knowledge of the model, with direct experience with the environment.

In order to obtain training data for the model, we also assume the availability of an oracle, which upon request can provide the agent with exact information about the current state. For example, in a spoken dialogue system, the agent can request the user to type in the word uttered, if it cannot be understood correctly. The state information is used only to improve the model, not in the action selection process. This means that there can be delays between the query request and the query processing. The answer to the query can also be noisy. The parameters of the Dirichlet distribution can also be updated directly from experience, without querying the oracle.

The algorithm is as follows. First, we assume that initial Dirichlet parameters are given, representing both a priori knowledge of the model, and uncertainty over model parameters. If appropriate, several parameters can share the same Dirichlet distribution (and hence use the same set of hyperparameters). Next, our agent samples a number of POMDP models according to this Dirichlet distribution. The agent then computes an (approximately) optimal policy for each of these models. At each time step, one of the models is chosen at random (with probability equal to the weight of each model under the current Dirichlet distribution) and the corresponding optimal action is applied. This reasonable execution performance throughout the active learning process. It also allows the agent to focus the active learning in regions of the state space most often visited by good policies. Each time an action is taken and an observation is received, the agent can decide to query the oracle for the true identity of the hidden state. If a query os performed, the Dirichlet distributions are updated according to its outcome. We note that the result of the query is used only to update the model, and has no impact on the action choices. This is crucial, as we assume that the agent will not be allowed to query after the learning process has been completed. Hence, the policy learned must be able to rely on the model alone.

In order to reduce as much as possible the number of queries, the agent can decide to use just the information from the action-observation sequence so far and the knowledge

obtained from previous queries in order to update the Dirichlet parameters. This is called *non-query learning*. In order to obtain the best information possible from the queries, the agent maintains an additional belief state associated with each model, called alternative belief. When a query is made, the alternative belief is set to reflect the full state information. Afterwards, it is updated based on the action-observation sequence in the standard Bayesian way. The alternative belief state is not used in planning; its role is to keep track of the information available from the previous query.

The decision on whether to perform a query or not is based on three factors:

- The variance in the values predicted for the best action among the current models; this is a very good indicator of how much learning remains to be done.
- The information gain of a query; this is computed based on the alternate belief
- The entropy in the mean alternate belief; if the entropy is high, a lot of knowledge has been lost since the last query, and a new query is in order.

In the current implementation, these are combined heuristically. MEDUSA can also adapt to non-stationarity in the environment, because discounting is used to weigh more recent experience.

MEDUSA scales nicely: one Dirichlet parameter is needed for each uncertain POMDP parameter, but the size of the underlying POMDP representation remains unchanged, which means that the complexity of the planning problem does not increase. However this approach requires the agent to repeatedly sample POMDPs from the Dirichlet distribution and solve the sampled models in order to select good queries.

We tested MEDUSA both on standard tasks from the POMDP repository, and recently, on a large robotic navigation task using the Carmen simulator. In all cases, MEDUSA converges to the correct model parameters in a reasonable number of queries, and the total number of queries stabilizes. This indicates that the learned policy does not rely on queries at all. More importantly, the performance is very good even during the learning process.

More detailed information on MEDUSA can be obtained from:

`http://www.cs.mcgill.ca/ rjaulm/`

References

[1] Anderson, B. and Moore, A. “Active Learning in HMMs”. ICML 2005.

[2] Cohn, D. A., Ghahramani, Z. and Jordan, M. I. “Active Learning with Statistical Models”. NIPS 1996.

[3] Dearden, R.,Friedman, N.,Andre, N., ”Model Based Bayesian Exploration”. UAI 1999.

[4] Jaulmes, R., Pineau, J., Precup, D. “Active learning in Partially Observable Markov Decision Processes”. ECML 2005.

[5] Jaulmes, R., Pineau, J., Precup, D. “Learning in non-stationary Partially Observable Markov Decision Processes”. ECML 2005 Workshop on learning in non-stationary environments.