EPSRC Engineering and Physical Sciences Research Council

BAE SYSTEMS

Bayesian Learning for Agent Coordination

M. Allen-Williams
Intelligence, Agents, Multimedia
School of Electronics and Computer Science
University of Southampton

Summary

In the example problem to the right, agents must learn while acting; learn about other agents while coordinating; and form models from partial data.

We examine a principled approach to such problems which extends Bayesian learning techniques into such partially observable domains. This provides us with a principled model-based approach having all the advantages of model-based methods such reusability and separability of individual agent models.

Extending this approach into our difficult domain necessitates the use of several efficiency techniques---some tried and tested in related work, such as the use of repair sampling or statistical clustering. To these established techniques we add the novel approach of using graphical inference techniques to perform updates to models in the agents' belief state at each step, passing messages through the hidden variables.

Example problem

There has been an earthquake. A number of ambulances, perhaps from different districts with different training practices, are dispatched to carry out the search-and-rescue operation.

- The situation is initially unknown
- Ambulances are not sure how the situation will evolve:
  unknown transition function
- Ambulances are not certain of how the others will behave:
  unknown strategies for the other agents
- Ambulances cannot observe the full situation at any one time:
  partial observability of states
- Ambulances are not always aware of what the others are doing:
  partial observability of actions
- There is a common, known goal (to rescue as many people as possible):
  co-operative system ; deterministic rewards
- Ambulances may broadcast news of rescues:
  full observability of rewards

In such a scenario, agents (ambulances) must try and cooperate to optimise the common reward, learning about the situation and about each other as they go. They may save this knowledge to try and apply it to future related scenarios.

Approach: Bayesian learning

- A form of *reinforcement learning*: agents update world models at each time step, based on their observations
- *Model-based:* an agent's learning process learns the environmental dynamics and the behavioural models of the other agents, and deduces the optimal action from these models
  - ***Why?***
  - Can separate problem variables
  - May be able to re-use parts of the model in future
  - May be able to make use of simulation steps if time steps are long
  - Principled
- Uses *Bayesian probabilities* over all models: the agent maintains a *belief state* which encapsulates the probability that any particular dynamic function is the true function (or pdf). A distribution form(e.g. Gaussian) is assumed for each relevant pdf, and the belief state holds probabilities over each variable associated with that form.
  Bayesian updates at each step using Bayes' rule:
  P(Model | Observations) = z.P(Observations | Model) P(Model)

- **Hidden variables:** have to be summed over. In our scenario where actions or states may be only partially observable, this may result in a chain of summations.
- Furthermore, variables which are independent may become dependent. For example, models of transition dynamics and models of agent behaviour are independent in a fully observed system. However, when actions are not observed, these two models become dependent.

.

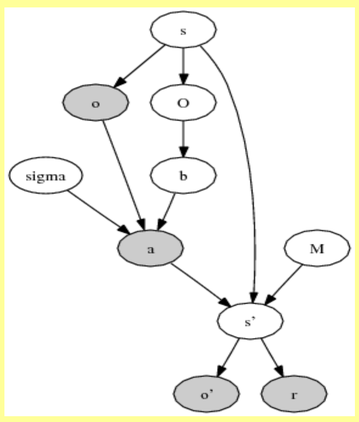
Bayesian updates in a graphical model

Rather than perform summations longhand, we make use of the message passing techniques of graphical models.

The figure on the rights shows an example Bayesian network for a system with partially observable states. The network represents a single step in the system. The aim is to compute the marginals P(M) and P(sigma), given the observed (gray) variables, and summing over the hidden (white) variables.

Observed: the agent's observations arising from the current and resulting states (o and o'), the actions of all the agents (a), and the reward from the resulting state.

Hidden: the current and resulting states (s and s'), the other agents' observations arising from the current state (O), and their resulting beliefs (b).



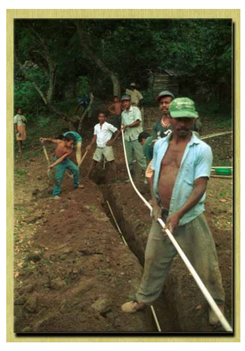
- We use the **junction tree** message passing technique
- We muse supply conditional probabilities on each node (or priors on the root nodes)
- Messages travel between the roots and the leaves (consider sharing information via a telephone tree)

•So far exact junction tree methods are sufficient, but there is potential to use faster approximate message passing algorithms on larger problems.

Efficiency Issues

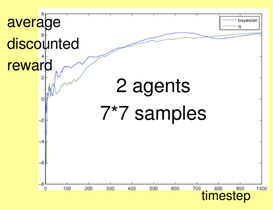
Various techniques to improve efficiency:

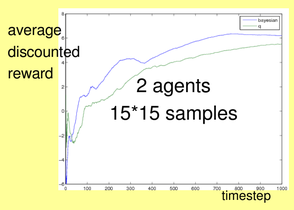
- *Sparse priors* to encode information about infeasible transitions
- *Repair-sampling* to speed up model updates
- *Statistical clustering* to merge similar states

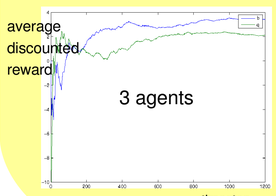


laying pipes

Results







The *sample size* refers to the number of samples taken when estimating an integral (i.e. summing over a continuous variable). The multiplication occurs in taking a double integral. We show, for a two-agent problem, that increasing the sample size makes the

Bayesian learner more effective.

Conclusions

We have demonstrated a method for coordinated reinforcement learning in multiagent systems and showed how it can be used in partially observable systems, exploiting the graphical structure of a Bayesian network to perform efficient model updates. Our system is comparable with a Q-learner on the same problem, and has the advantages of model-based systems.

Future Work:

- Use hierarchies to carry out variable abstractions
- Exploit independencies between variables such as individual state variables
- Make use of rewards in inference
- Unknown or partially observable rewards
- Malicious agents...

A smaller problem:

A multi-lingual team of rescue workers tries to dig a trench in which a pipe carrying an emergency water supply will lay flat, coordinating to make the depth consistent:
the trench should be below ground level so the pipe can be covered
digging too deep will let groundwater into the trench causing digging problems
post-earthquake there may be unexpected behaviour such as further landslips
partially observable:
the team may be able to look at the level of the pipe, but not see one another
the team may be able to see one another, but not the level of the pipe (as in the underwater pipe in the picture to the right)

Timing

In our small example, the number of states is exponential in the number of agents. As we move from two agents to three agents, therefore, the system becomes exponentialy slower. In practice we found that for a two-agent problem our system was approximately half as fast as the Q-learner, taking a fraction of a second for each step. For a three-agent problem our system was noticeably slower than the Q-learner, but still managed to compute an action for each of the three agents in approximately 0.7s per agent. Further speedups are possible, as well as a more efficient implementation (we used MATLAB).

References

- Chalkiadakis, G and Boutilier, C. (2003). Coordination in Multiagent Reinforcement learning:a Bayesian approach. In Autonomous Agents and Multiagent Systems (AAMAS 2003).
- Dearden, R., Friedman, N., and Andre, D. (1999). Model- based Bayesian exploration. In *Uncertainty in Artificial Intelligence (UAI-99),*

ALADDIN autonomous learning agents for decentralised data and information networks

ECS Electronics & Computer Science University of Southampton