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.

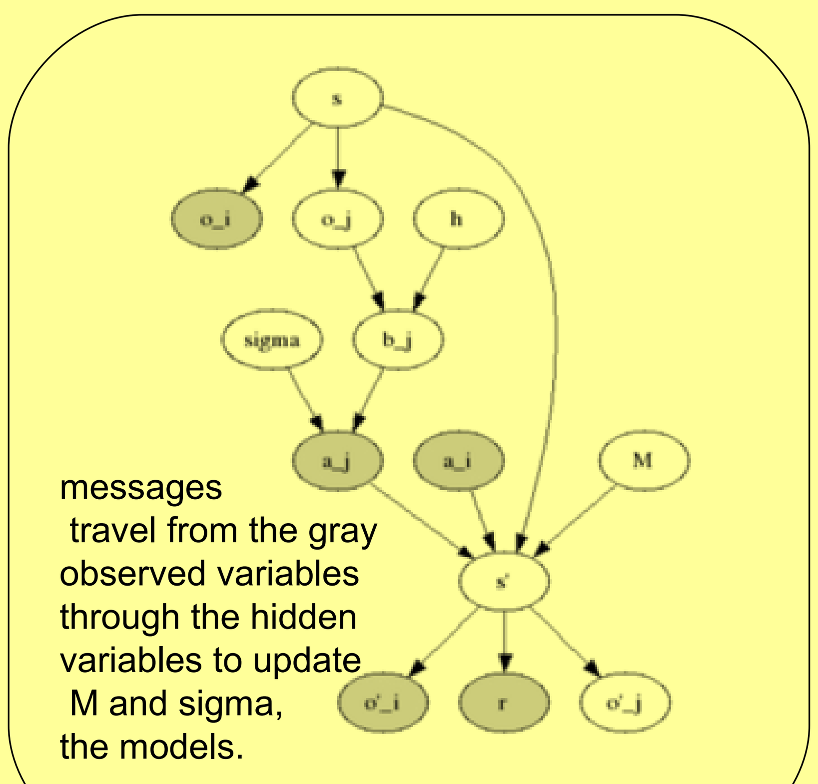
Our approach

Use Bayesian learning:

- model based learning:
  - learn models not just how to act
  - reuse models in future situations
  - separate problem variables
- maintain beliefs over all possible models
- principled approach

Efficiency issues:

- *sparse priors* encode information about infeasible transitions
- *repair-sampling* speeds up model updates



Bayesian Updates

More efficiency:

- Exploit *graphical structure* for updates (see figure)
  - use graphical inference mechanisms for complex updates: *junction tree* message passing algorithm
  - permit use of approximate techniques in large systems
- *Statistical clustering* to merge related states
  - if states lead to the same action, they are functionally equivalent
  - statistical clustering is easy to update incrementally
  - easy to merge or create clusters
  - binary “features” to exploit variables in states

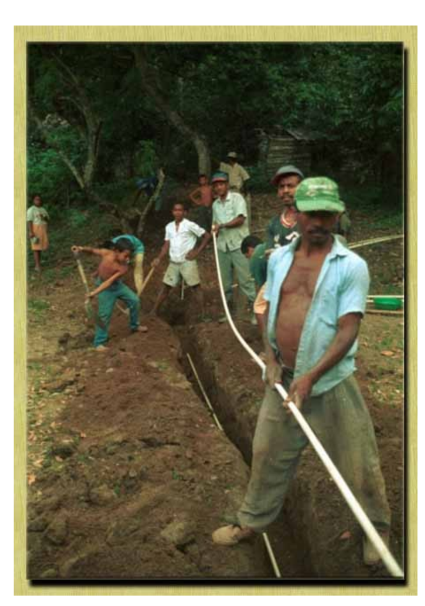
Future:

- *Hierarchical* state/action clustering
- Exploit dependencies between individual state variables
- 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:

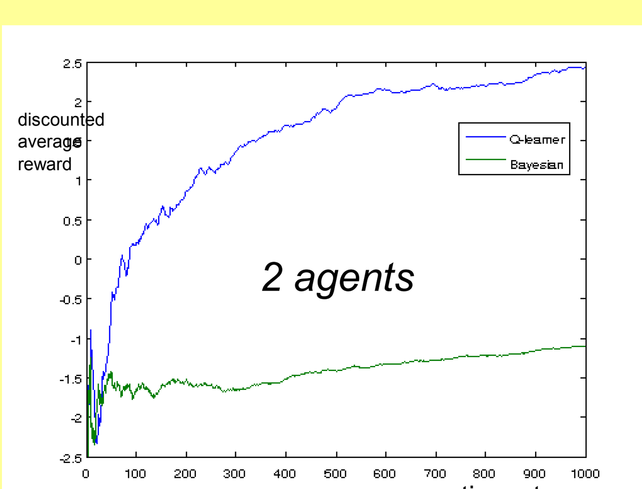
- the trench should be below ground level so the pipe can be covered
- digging too deep will let groundwater into the trench causing digging problems
- post-earthquake there may be unexpected behaviour such as further landslips
- *partially observable:*
  - the team may be able to look at the level of 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)

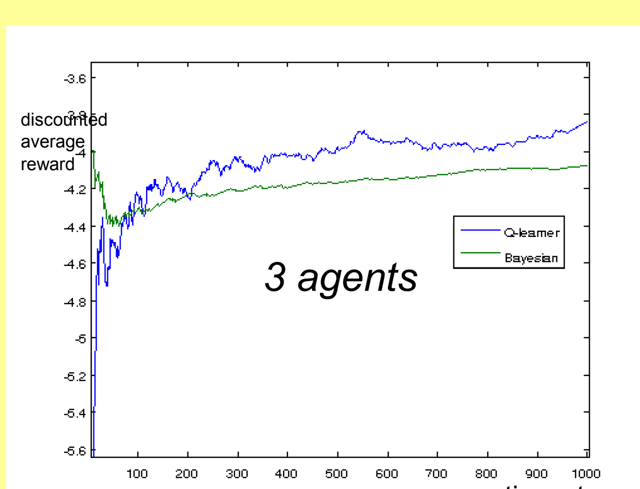


laying pipes

Results:

We compare the Bayesian model learner with a simple Q-learner for problems with two and three agents.





Timing: with 3 agents, each agent takes ~0.7s to update and act.

As the number of agents increases, the benefits of learning the separate models begin to kick in.

ALADDIN autonomous learning agents for decentralised data and information networks

ECS Electronics & Computer Science University of Southampton