Reinforcement learning for spoken dialog systems:

Using POMDPs for Dialog Management

Steve Young

Cambridge University Engineering Department
Machine Intelligence Laboratory

Outline of Talk

- the promise of statistical dialog systems
- Markov Decision Processes and their limitations
- Partially Observable MDPs – an intractable solution?
- the Hidden Information State system – a proof of concept.

Statistical Dialog Systems

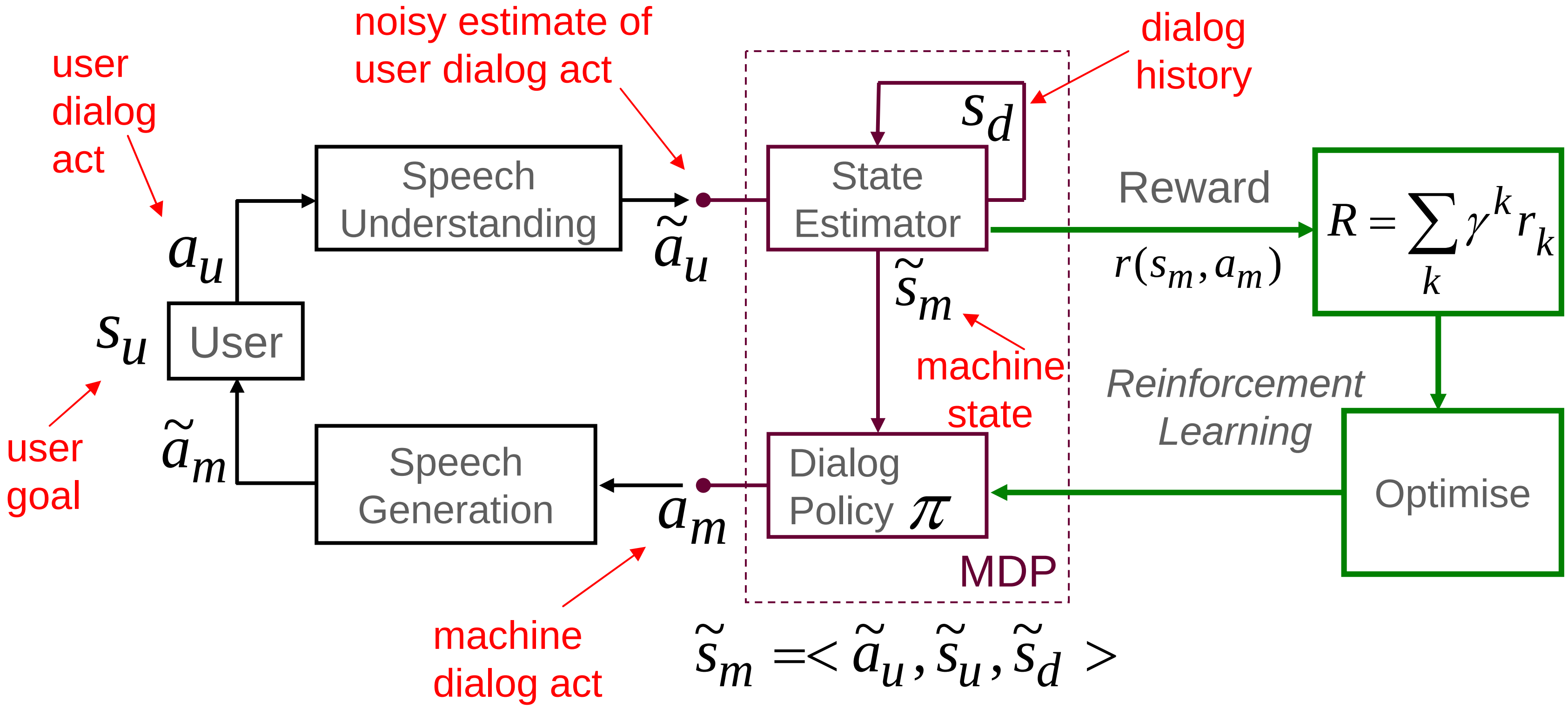
A statistical approach to dialog system design offers the following potential advantages:

- ❑ formalise dialog design criteria as objective reward functions
- ❑ automatically learn dialog strategies from data
- ❑ allow decision making to be optimised
- ❑ increase robustness to recognition/understanding errors
- ❑ enable on-line dialog policy adaptation to allow the system to learn from experience

Overall, increase robustness and reduce design, implementation and maintenance costs

Markov Decision Processes provide the framework to do this .....

Dialog as a Markov Decision Process



Levin, E. and R. Pieraccini (1997). "A Stochastic Model Of Computer-Human Interaction For Learning Dialog Strategies." Proc Eurospeech, Rhodes,Greece.

Levin, E., R. Pieraccini, et al. (1998). "Using Markov Decision Processes For Learning Dialog Strategies." Proc Int Conf Acoustics, Speech and Signal Processing, Seattle,USA.

Training an MDP

Key idea is to associate a value function with each state

$$V^{\pi}(s_m) = E_{\pi}\{R \mid s_m\}$$

$$Q^{\pi}(s_m, a_m) = E_{\pi}\{R \mid s_m, a_m\}$$

where $Q^{\pi}(s_m, \pi(s_m)) = V^{\pi}(s_m)$

Given V or Q, policy optimisation is straightforward since if

$$Q^{\pi}(s_m, \pi'(s_m)) > V^{\pi}(s_m)$$

then policy π' is better than policy π .

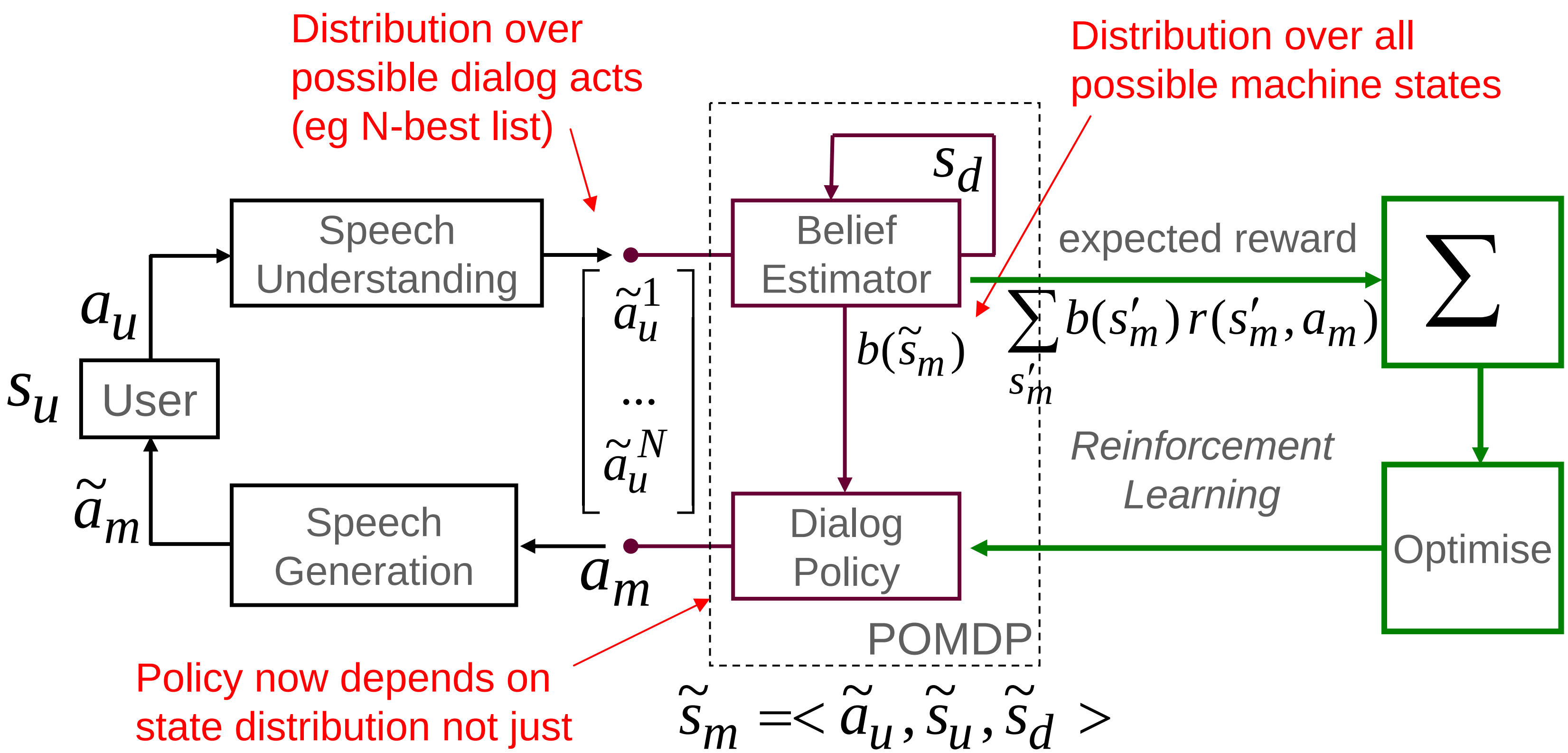
A popular algorithm for implementing this is ***Q-Learning***

Limitations of MDP Framework

Modelling dialog as an MDP suffers from a variety of practical problems

- state space is huge, hence propositional content and much of the relevant history is often ignored.
- dialogs are fragile because user state s_u and user dialog act a_u are uncertain, hence estimate of machine state $\tilde{s}_m$ is often incorrect
- recovery strategies are difficult since no information is available for backtracking
- no principled way to handle N-best ASR output.

Dialog as a Partially Observable MDP



Roy, N., J. Pineau, et al. (2000). "Spoken Dialog Management Using Probabilistic Reasoning." Proceedings of the ACL 2000.

Williams, J., P. Poupart, et al. (2005). "Factored Partially Observable Markov Decision Processes for Dialog Management." 4th Workshop on Knowledge and Reasoning in Practical Dialog Systems, Edinburgh.

Belief Update Equation

Belief is updated every dialog turn as follows:

Observation Model | User Action Model | Transition Model

$$b'(s'_m) = k.P(o' \mid a'_u)P(a'_u \mid s'_m, a_m)\sum_{s_m} P(s'_m \mid s_m, a_m).b(s_m)$$

new belief

observed recogniser output (can include multiple hyps)

hypothesised user dialog act

state transition network

old belief

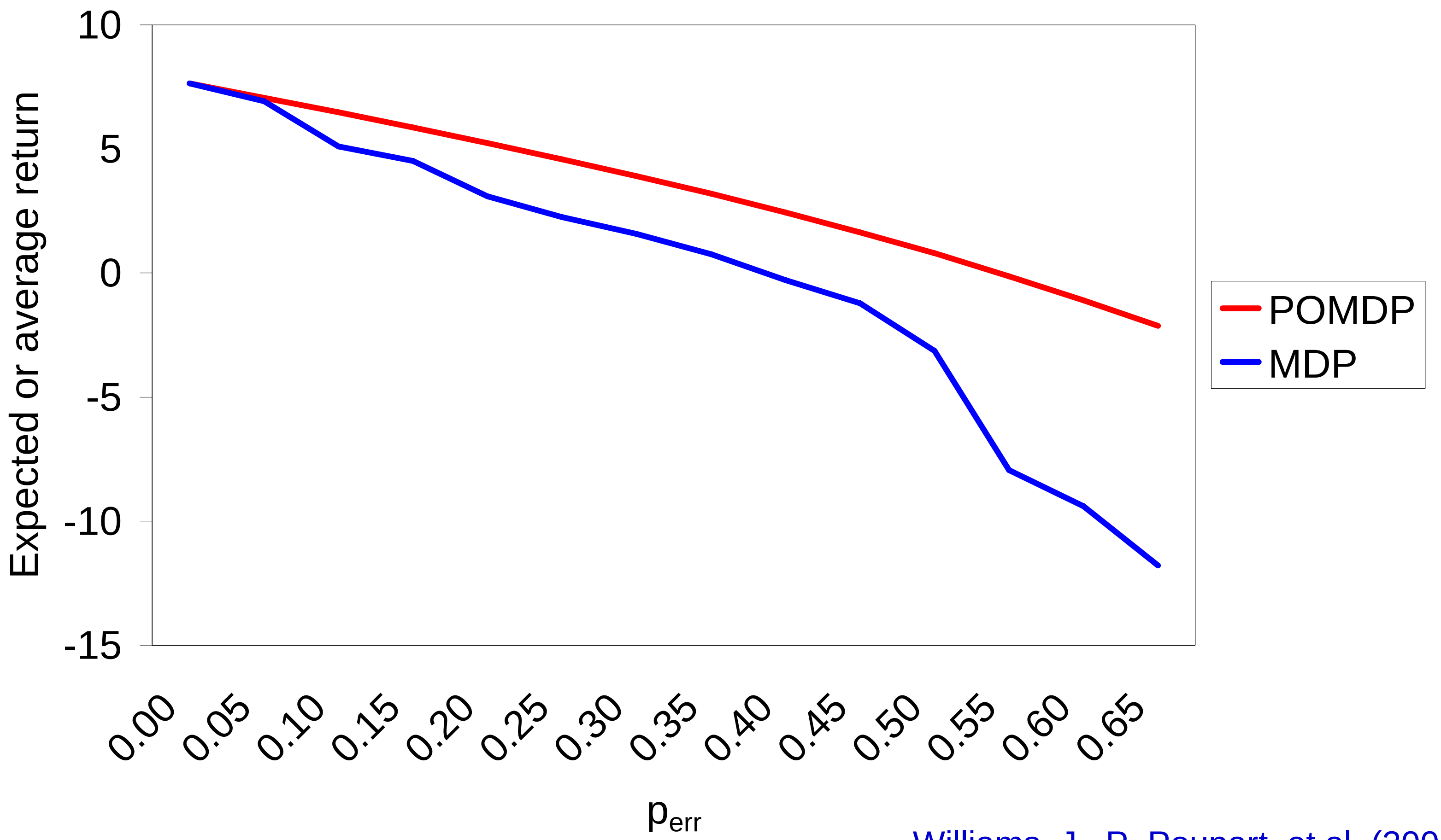
new evidence

prior of a_u given s'_m and a_m

probability of s'_m

Robustness of POMDP vs. MDP

Simulation of simple 2 slot 3-city travel problem



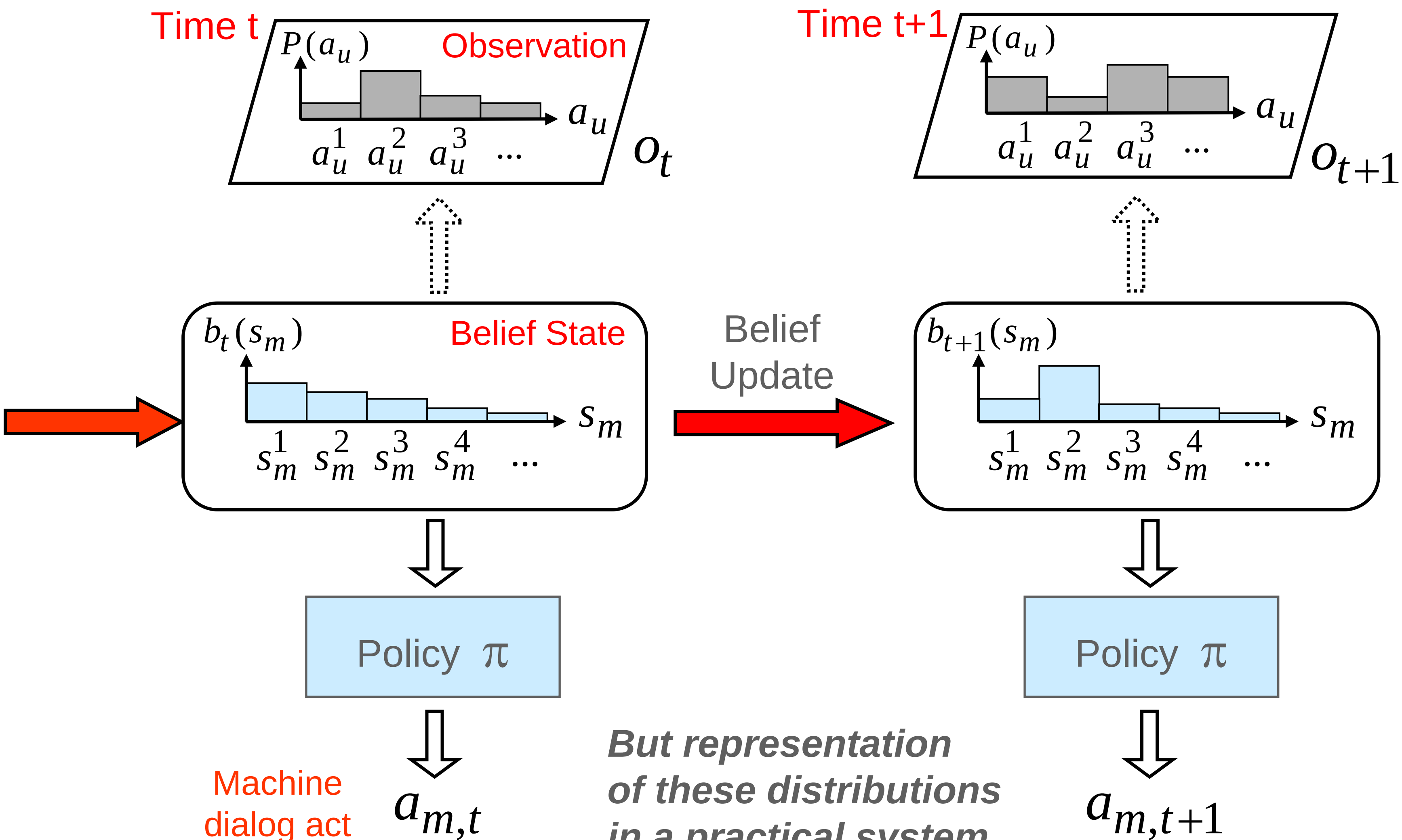
Williams, J., P. Poupart, et al. (2005).

Summary of the POMDP Framework

- system maintains multiple dialog hypotheses called the *belief state*
- machine actions are based on the full belief state distribution not just the most likely state
- no backtracking is required when misunderstanding detected
- speech understanding output is regarded as an *observation*
- belief distribution is re-computed each time a new observation is received in a process called *belief monitoring*
- N-best ASRU outputs naturally incorporated into belief monitoring framework via an *observation model*
- POMDP framework naturally includes a *user model* which gives probability of each user act given each possible dialog hypothesis

However, there are some issues ....

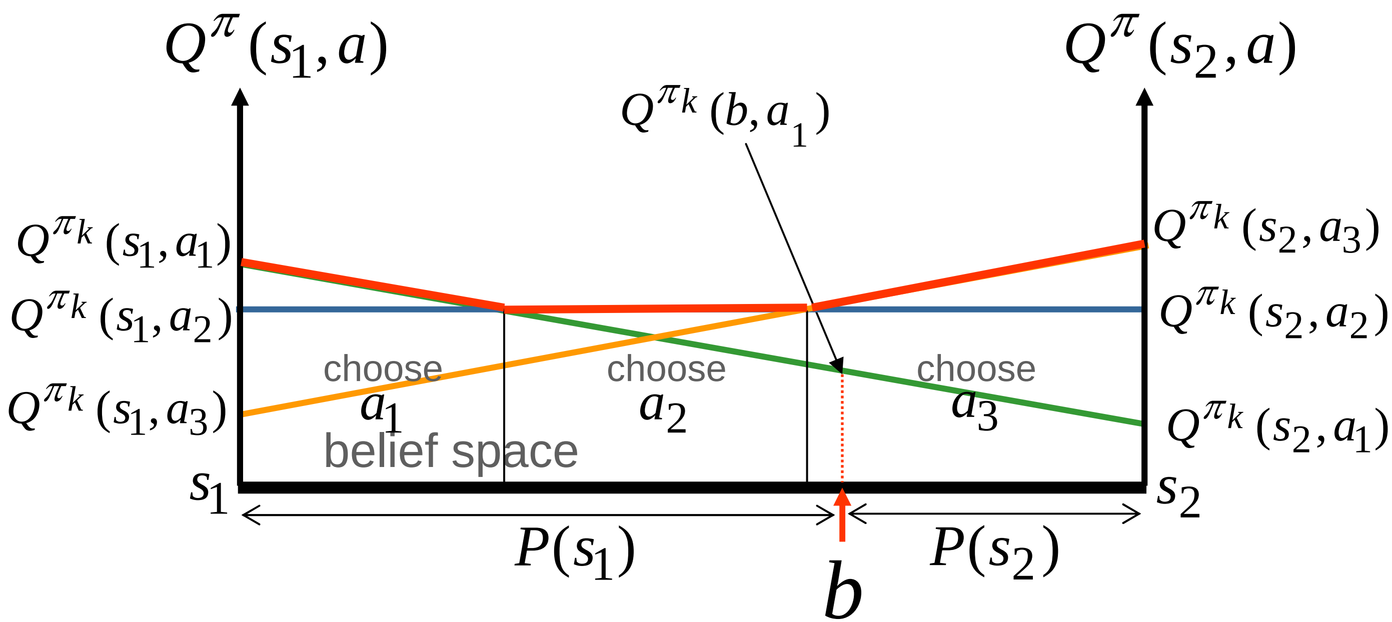
Belief Monitoring



But representation of these distributions in a practical system is unclear.

POMDP Value functions

Consider a system with just two states and 3 actions



POMDP value functions are hyperplanes in belief space.

Upper surface provides defines the value function *V(b)*.

Exact learning is iterative and ***effectively intractable.***

Scaling to Real Systems

- POMDPs provide an elegant mathematical framework for modelling spoken dialog systems but ....
- State space will be huge – direct belief monitoring is impractical.
- Exact POMDP optimisation is intractable - even approximate POMDP optimisation is limited to a few thousand states

A solution – the *Hidden Information State* Dialog Model

The Hidden Information State Model

The HIS model provides a scaleable POMDP framework for implementing practical spoken dialog systems.

- ❑ Partition state space and compute partition beliefs *not* state beliefs
- ❑ Represent user goals by branching-tree driven by ontology rules.
- ❑ Maintain two state spaces: master space and summary space. Monitor beliefs in master space, apply and optimise policies in summary space
- ❑ Use grid-based approximations, hence finite policy table

Structure of a HIS Dialog Hypothesis

$$h = < \{s_u\}, a_u, s_d >$$

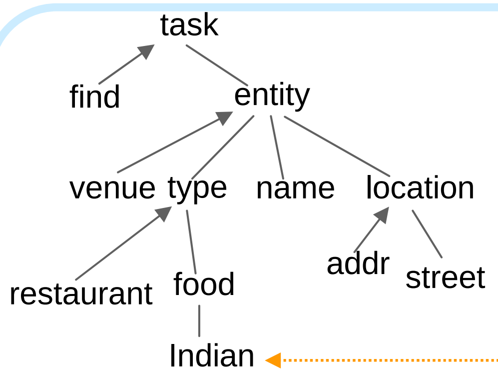
ie a set of s_m with common a_u & s_d

a partition of s_u

User goal – tree structured set of entities

User action – a dialog type plus goal tree bindings

Dialog history – grounding status of each tree node



inform(food=Indian)

restaurant :	UserRequested
food :	UserInformed
area :	Grounded
name :	Initial

A single hypothesised information state

User goal tree built incrementally from rules, expanded on demand to accommodate user dialog acts

HIS Partitions

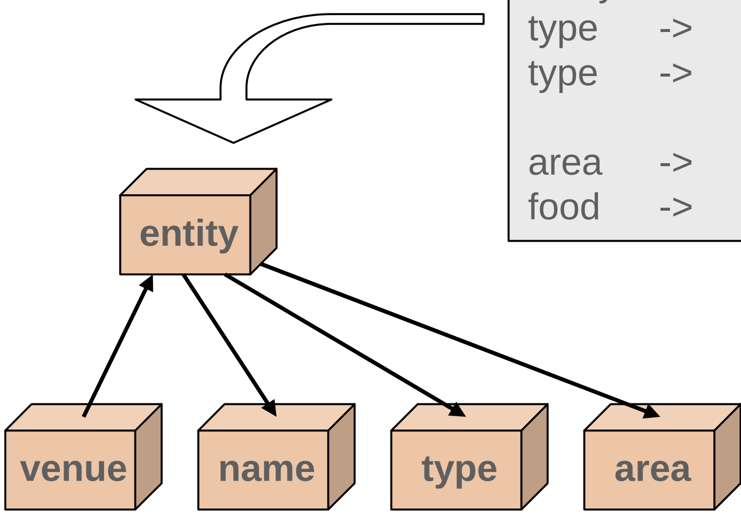
- Each partition represents a group of user goal states
- Partitions are stored as tree structures, with nodes defined by a task ontology
- Partitions are split by incoming user dialog acts
- When a partition is split, its belief is shared between the splits

Example ontology rules

entity	->	venue(name,type,area)	1.0
type	->	bar(drinks,music)	0.4
type	->	restaurant(food,price)	0.3
area	->	(central | east | west | ....)	
food	->	(Italian | Chinese | ....)	

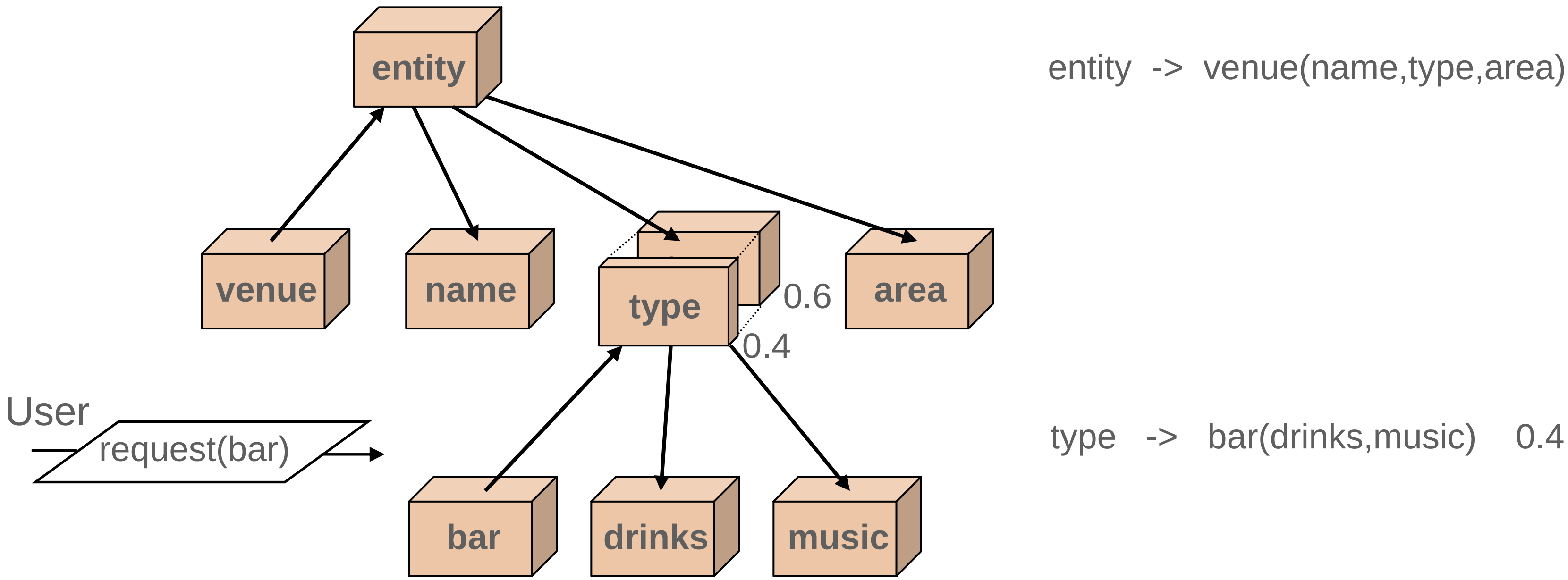
Structure rules with prior probs

Lexical/Dbase rules



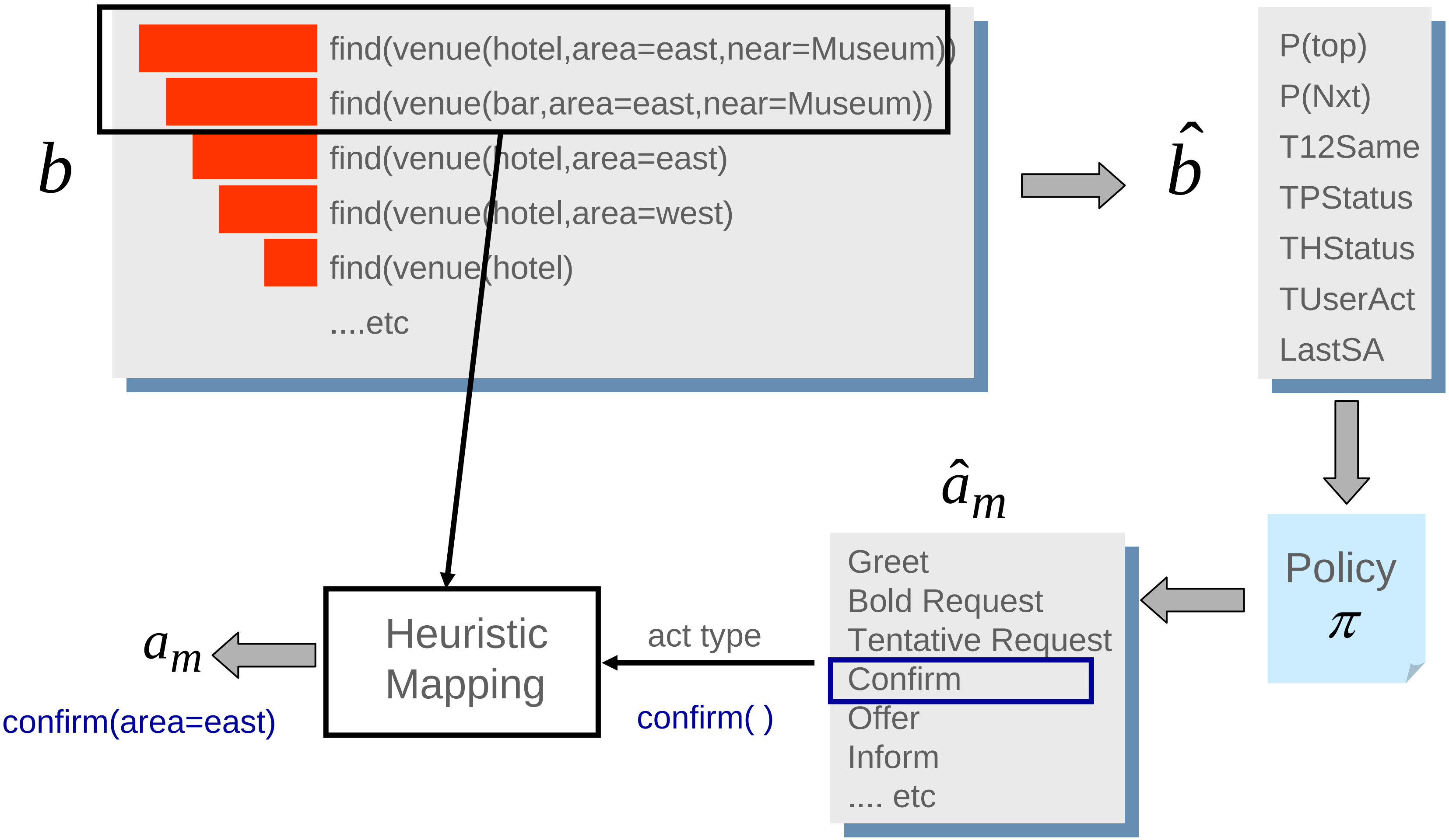
Partition splitting

- Incoming dialog acts cause partitions to be extended and split in order to match the items in the dialog act with the nodes in the tree.



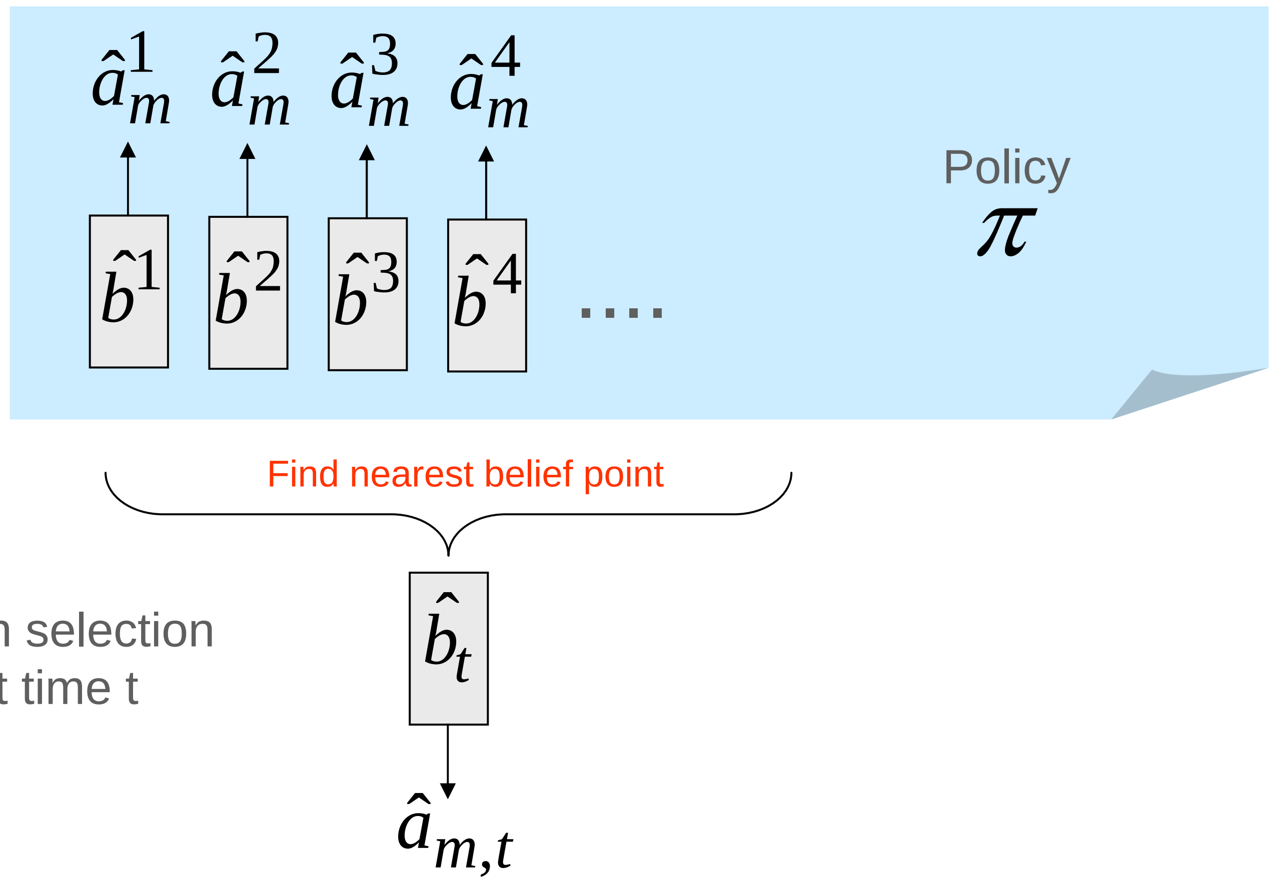
Master <-> Summary State Mapping

Master space is mapped into a reduced summary space:



The POMDP Policy and Action Selection

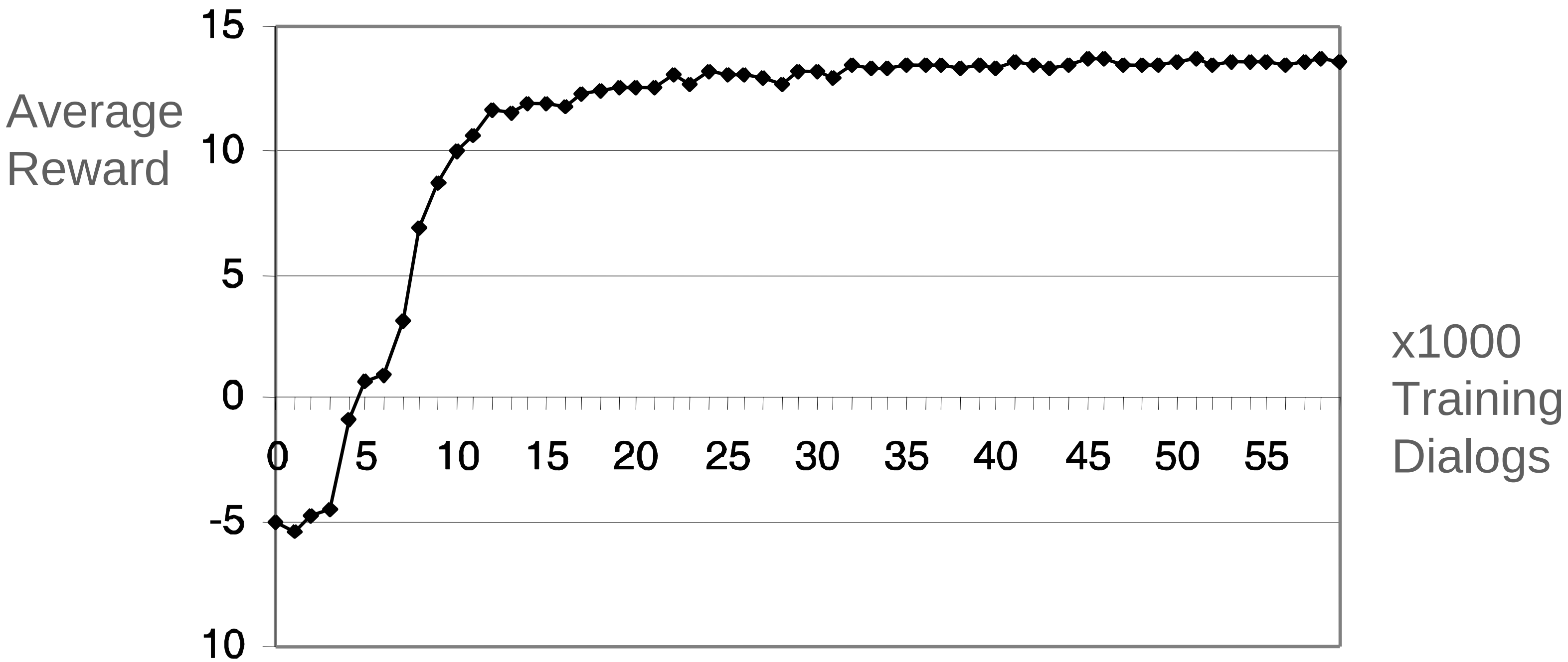
A set of points in summary space and their associated actions



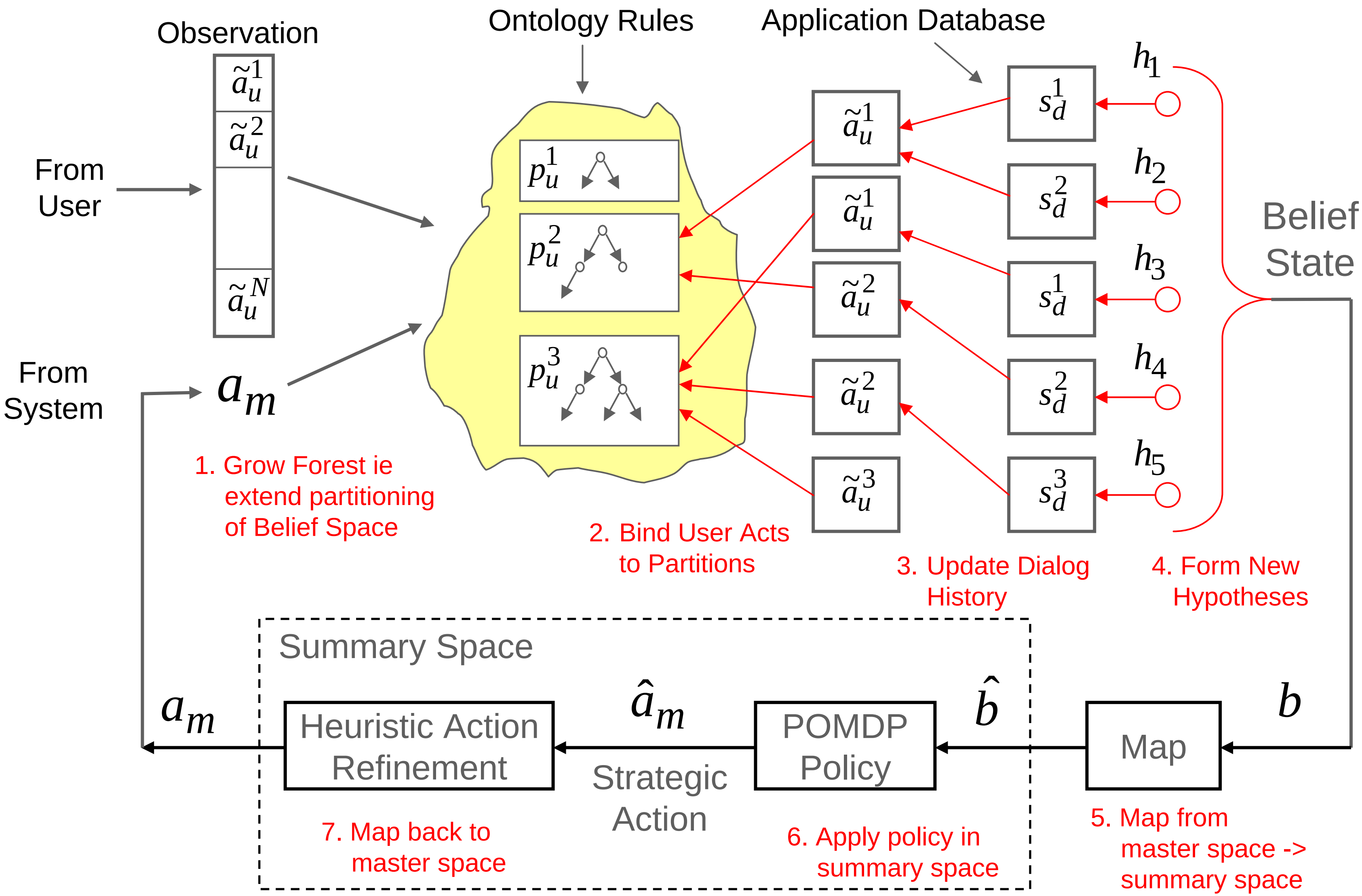
Action selection
at time t

Policy Optimisation

- Use Q-learning with a simulated user on belief points
- Start with a single belief point
- Add new points as they are encountered upto some maximum



Summary of HIS Dialog Manager Operation



Evaluation

The system was tested by human users in a two day study conducted simultaneously at Edinburgh and Cambridge.

Dialogues were deemed to be successfully completed when the system made a correct recommendation.

	Cambridge	Edinburgh	Combined
# subjects	23	17	40
# dialogues	92	68	160
% WER	21.1	37.3	29.3
% completion rate	95.7	83.8	90.6
Average turns to completion	3.8	8.1	5.6

Work supported by the EU FP6 “TALK” Project

Conclusions

- Partially observable MDPs provide a natural framework for modelling spoken dialog systems:
  - explicit representation of uncertainty
  - support for N-best ASR output
  - incorporates user and observation model
  - simple error recovery by shifting belief to alternative hypotheses
  - potential for on-line adaptation
- The Hidden Information State system demonstrates that POMDPs can be scaled to handle real world tasks
- There are many issues to resolve e.g. effective observation and user models, choice of summary state mapping, improved training procedures ...
- ... but overall POMDPs provide an opportunity for making significant improvements to both the design and implementation of spoken dialog systems.