NPFL099 - Statistical dialogue systems

Dialogue management

Action selection I

Filip Jurčíček

Institute of Formal and Applied Linguistics
Charles University in Prague
Czech Republic

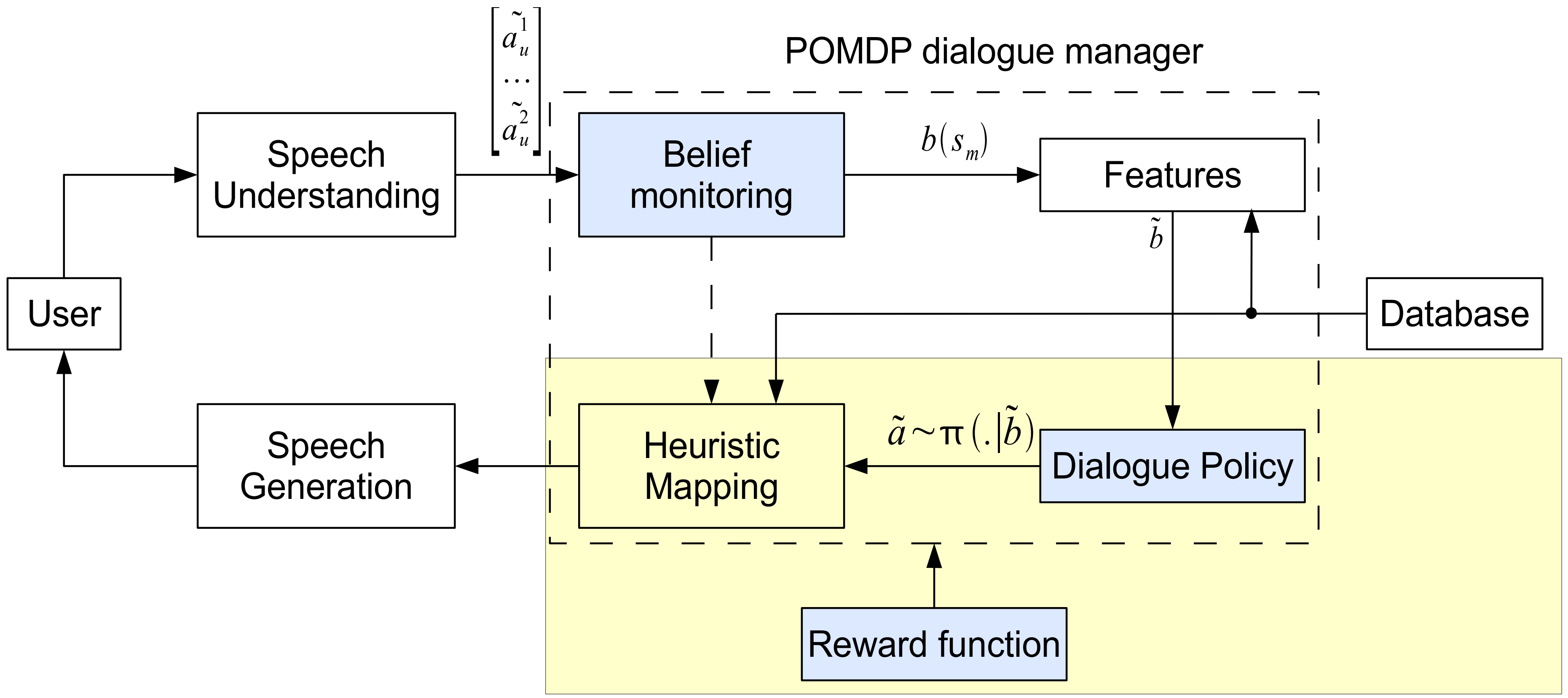
Home page: http://ufal.mff.cuni.cz/~jurcicek

Version: 02/04/2013

Outline

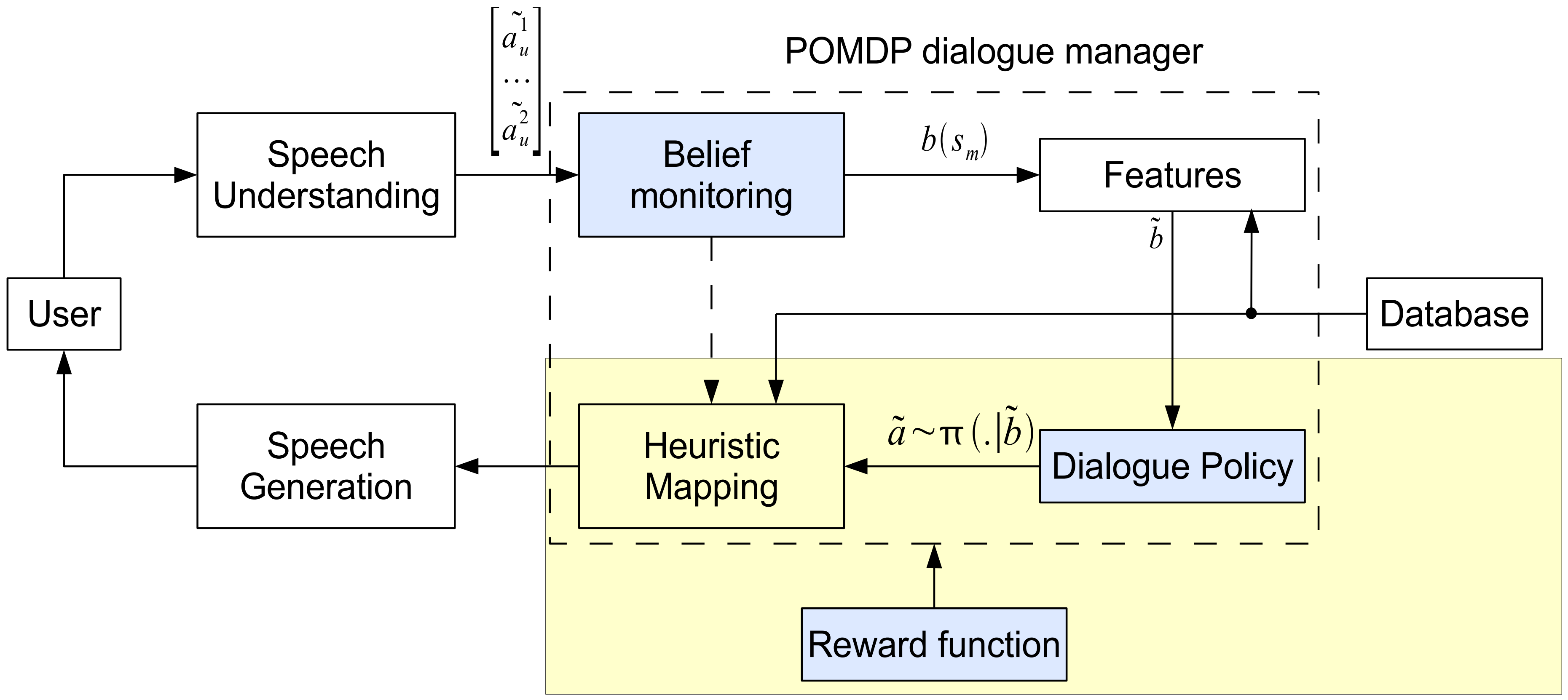
- Action selection in dialogue systems
- Why not supervised learning?
- Summary space
- Reinforcement learning
  - Model based RL
  - On-line learning

Action selection



- Actions are selected based on the uncertainty in b(s)
- Some actions can lower uncertainty in b(s)

Action selection



- Ideally the policy is optimized to maximize the reward function

Use of supervised learning

- Having a corpus of dialogues and optimal actions
  - one can use supervised learning to learn what to do

- Problems
  - User has a much better
    - speech recognition
    - understanding
    - model of the environment
    - actions and a model of their use
  - Some times it is not clear what to do / or what is the best
    - e.g. one can continue in a dialogue in two, equally good, ways

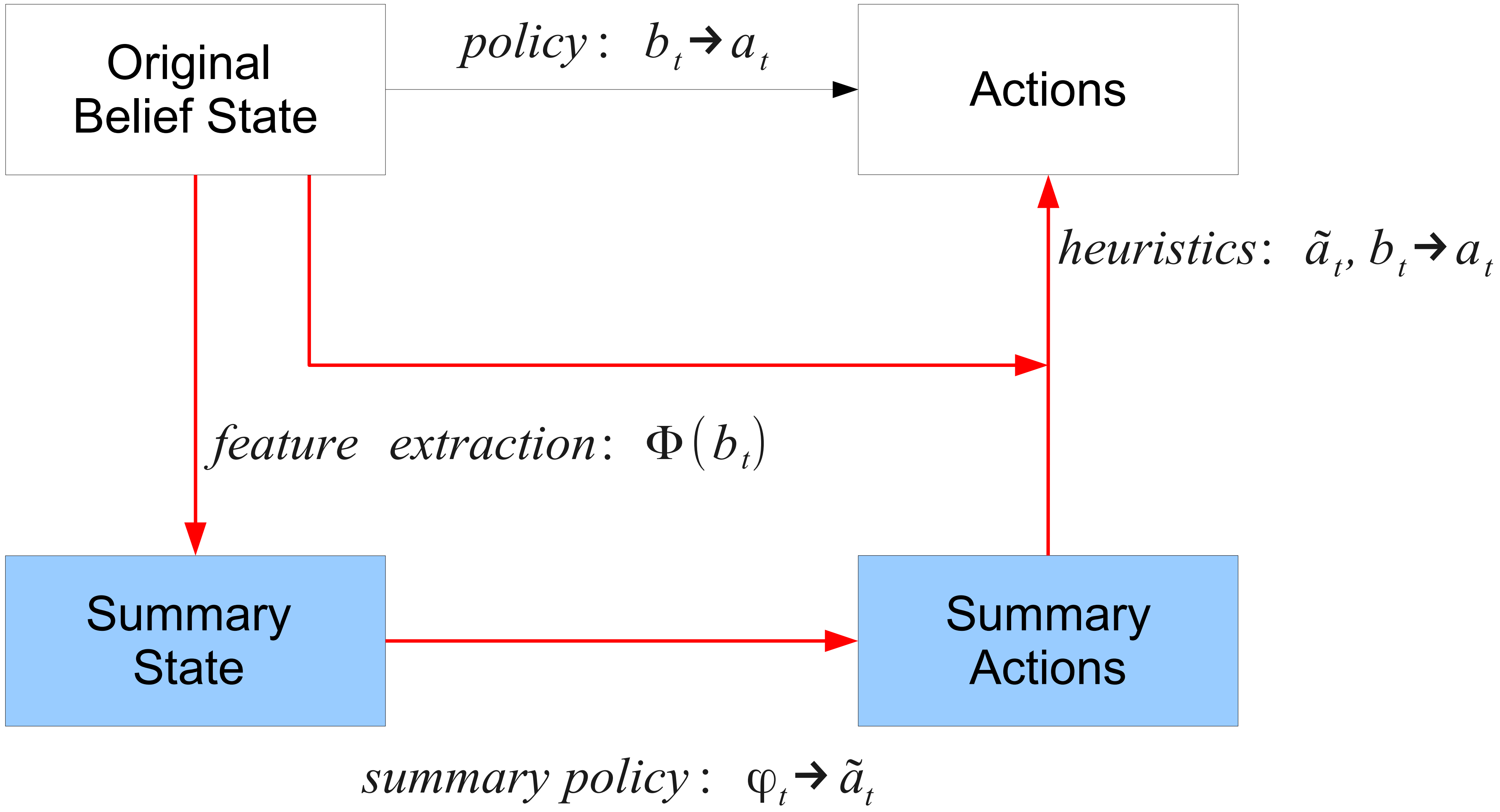
Use of supervised learning

- Don't learn exactly what to do
  - a user can randomly choose between two optimal actions
- Instead, learn to be as good as humans or even better than humans
- Learning exactly what humans are doing does not have to be the best thing to do

Dialogue policy learning

- In general, learning dialogue policies is too complex
- In a trivial solution
  - the complexity grows exponentially with the number of dialogue states and the actions
- Therefore, some approximations are needed
  - summary based policy

Summary based policy


Original Belief State
$policy:\ b_t \rightarrow a_t$
Actions
$heuristics:\ \tilde{a}_t, b_t \rightarrow a_t$
$feature\ \ extraction:\ \Phi(b_t)$
Summary State
Summary Actions
$summary\ policy:\ \varphi_t \rightarrow \tilde{a}_t$

Summary belief space

- When deciding what to do next
  - not all information from belief space is necessary

- Sometimes, the belief state can be summarised in the form of features

- However, we have to track the exact belief state

  - since all* information from the belief state is necessary for belief update

HIS Summary belief space

- Based on the two most probable partitions

- It includes
  - probability of the most likely partition
  - probability of the second most likely partition
  - status of the most likely partition
  - status of the hypothesis
  - grounding info
    - grounding info for each slot tracks whether:
      - user informed, user requested, system informed, system queried,
      - denied, grounded (acknowledged)
  - last user act

HIS Summary belief space

- Status of a partition

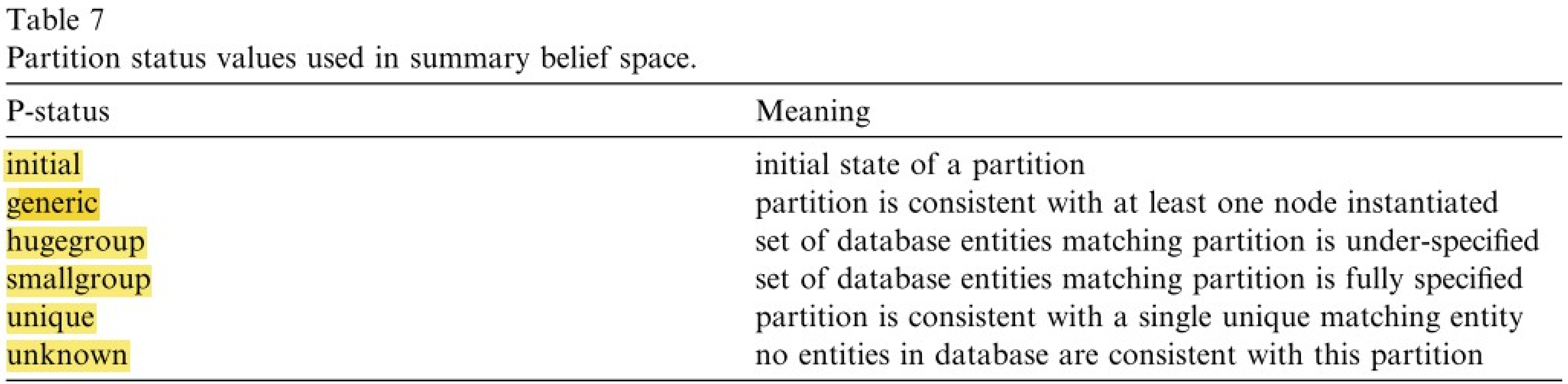


Table 7
Partition status values used in summary belief space.

P-status	Meaning
initial	initial state of a partition
generic	partition is consistent with at least one node instantiated
hugegroup	set of database entities matching partition is under-specified
smallgroup	set of database entities matching partition is fully specified
unique	partition is consistent with a single unique matching entity
unknown	no entities in database are consistent with this partition

- Status of a hypothesis

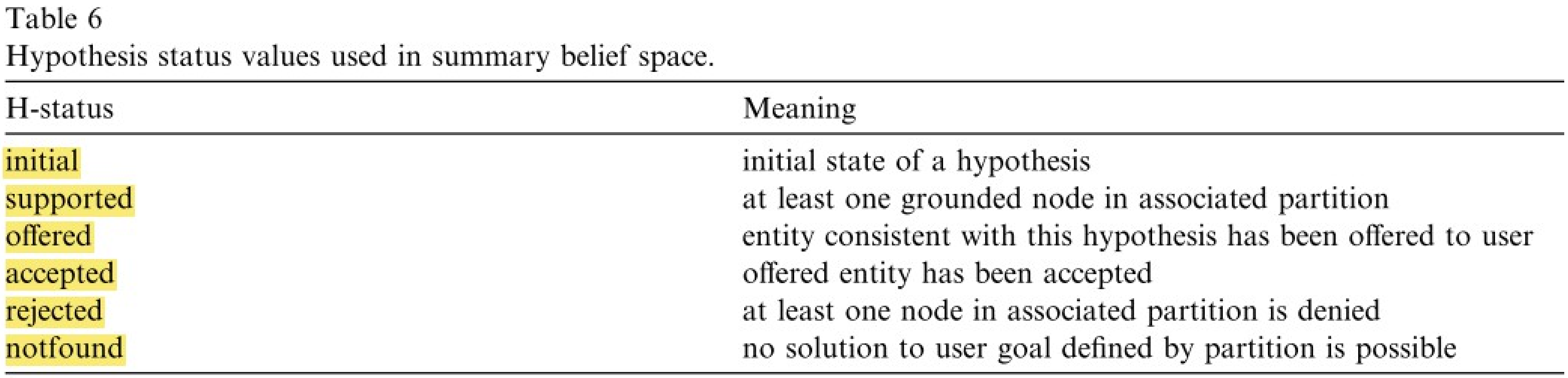


Table 6
Hypothesis status values used in summary belief space.

H-status	Meaning
initial	initial state of a hypothesis
supported	at least one grounded node in associated partition
offered	entity consistent with this hypothesis has been offered to user
accepted	offered entity has been accepted
rejected	at least one node in associated partition is denied
notfound	no solution to user goal defined by partition is possible

BUDS Summary belief space

- Slot level features (for each node in BN)
  - entropy in a node, and/or
  - probabilities of two most likely values in a node

- Global features
  - number of matching venues in a database
  - number of accepted nodes
  - probabilities of values in method and discourse nodes
    - the method node tracks the way user requests for venues (by constraints or by name)
    - the discourse node tracks whether user wants to finish, restart the dialogue or repeat the last system dialogue act

Summary actions

- Not all actions are applicable in every context
  - confirm only the most likely values in slots
  - select between the two most likely values
  - inform about the best matching venue
- The dialogue system can then work only with
  - prototypical (sometimes called summary) actions
- A heuristic is needed to convert summary actions into full dialogue acts

J. Williams and S. Young (2005). "Scaling up POMDPs for Dialogue Management: The Summary POMDP Method."

HIS Summary actions

- Small compact set of actions
- A lot of effort is send on the heuristic mapping the actions into the full dialogue acts

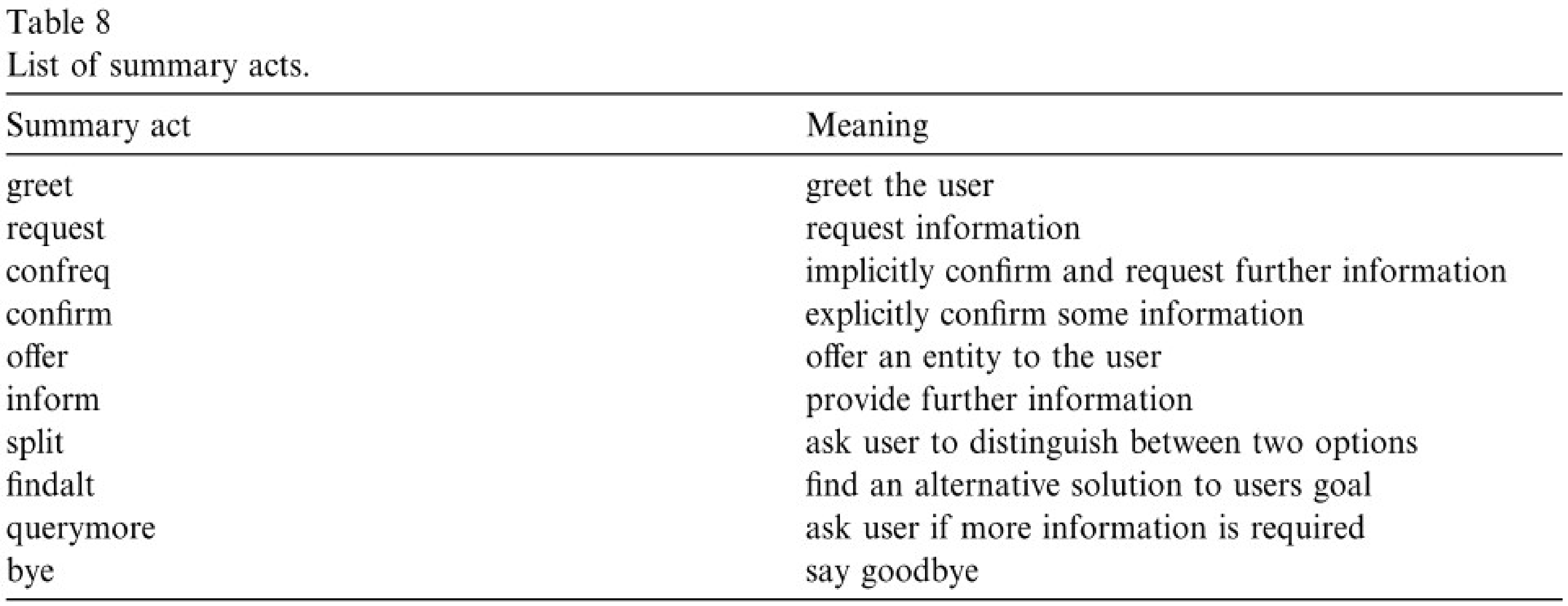


Table 8
List of summary acts.

Summary act	Meaning
greet	greet the user
request	request information
confreq	implicitly confirm and request further information
confirm	explicitly confirm some information
offer	offer an entity to the user
inform	provide further information
split	ask user to distinguish between two options
findalt	find an alternative solution to users goal
querymore	ask user if more information is required
bye	say goodbye

BUDS Summary actions

- Slot-level summary actions for each node*
  - request, confirm, select
- Global actions (similar to his)
  - inform about a venue
  - inform more
  - inform requested slots
  - inform slots being confirmed
  - inform about alternatives
  - inform by name
  - restart
  - repeat
  - request more
  - bye

HIS vs. BUDS

- Example TownInfo domain
- HIS
  - 10 summary actions
  - feature vector: 5 features
    - 2 continuous
    - 3 discrete
- BUDS
  - 10 global summary actions + 10 nodes * 3 slot-level actions
  - feature vector: 43 nodes * (3 or 8 features) + global features

HIS vs. BUDS

- BUDS summary space is more complex
  - Therefore, needs more training data to train
  - It could be potentially better tuned
- Although, the BUDS model is more detailed
  - users do not rate the BUDS system significantly better

Action selection

- Dialogue policy selects actions given the belief state
- Hand-crafted
  - in the form if / then statements
- Data driven
  - Deterministic - take an action a in summary belief state $\Phi(b_t)$
    $$\pi(b(.;\tau)) = \underset{a'}{argmax}\, Q^{\pi}(b(.;\tau), a';\theta)$$
  - Stochastic - sample an action from a distribution
    $$a \sim \pi(.|b_t) \approx \pi(.|\varphi(b_t))$$

Example of a handcrafted policy

1. request values for slots in which the most likely value has probability lower than 0.3
2. confirm the most likely values in slots where the most likely value has probability between 0.3 and 0.9
3. select between the two most likely values if the sum of the probabilities for these values is more than 0.8
4. inform about a venue if there is only one venue matching the provided constraints and the probability of the “byconstraint” value in the “method” node is more than 0.5
5. ...
6. …
7. …
8. …
9. ...
10. ...
11. provide more (the “inform more” summary action) information about the last offered venue if the probability of the “more” value in the “discourse” node is more than 0.5,
12. request more information from the user.

Action selection

- Data driven methods are based on reinforcement learning and the theory of Markov Decision Processes

- Two types
  - assuming that we know the model of a behaviour of the environment – the user

  - or we can learn directly from the observed behaviour

Reinforcement learning

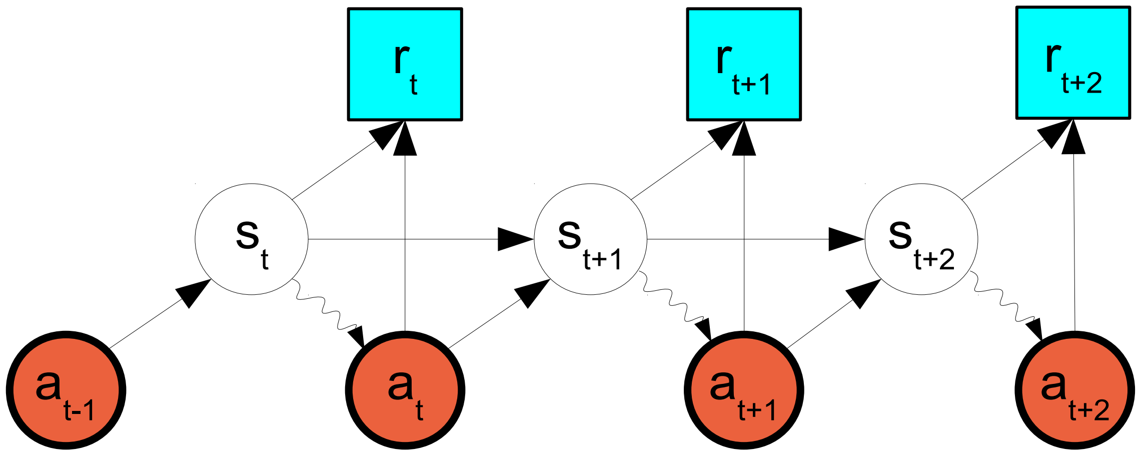
- A method for learning from its own errors
- Try some actions and observe the reward

Markov decision processes

- Markov decision processes - MDPs
  - model decision-making in situations where outcomes are
    - partly random and
    - partly under the control
  - attempt to optimise long term reward
    - the reward can be random too
- Reward can include:
  - number of turns
  - success in task
  - level of naturalness
  - subjective feeling of a dialogue being good

MDP Introduction

- MDP is defined on states, **s**, and actions, **a**
- For every action $\mathbf{a}_t$ taken in a state $\mathbf{s}_t$,
  - we receive reward $\mathbf{r(s_t,a_t)}$
- This repeats as long as necessary



- We want to take actions **a** in a way which will maximise

$$R=\sum_t r(s_t,a_t)$$

Model based reinforcement learning

- RL is maximising $R = \sum_t r(s_t, a_t)$
  - MB RL assumes knowledge of $P(s_{t+1} | s_t, a_t)$
  - model describes how actions affect the state.

- For example:
  - $P(s_{t+1} | s_t, a_t)$ - defines what is the chance that after asking for type of food the user is looking for, the user will answer the question
    - in essence it is our dialogue model
    - problem is that our dialogue model is crude
      - and it should not be used for this purpose
    - still we can try to use it

Model based reinforcement learning

- To get a dialogue policy one can define:
  - $Q(s,a)$ - expected future reward of $R = \sum_{t=1, s_1=s, a_1=a} r(s_t, a_t)$
    - taking action **a**
    - in state **s**

- A dialogue policy is then
  - $\pi(s) = \underset{a'}{argmax}\, Q^{\pi}(s, a')$

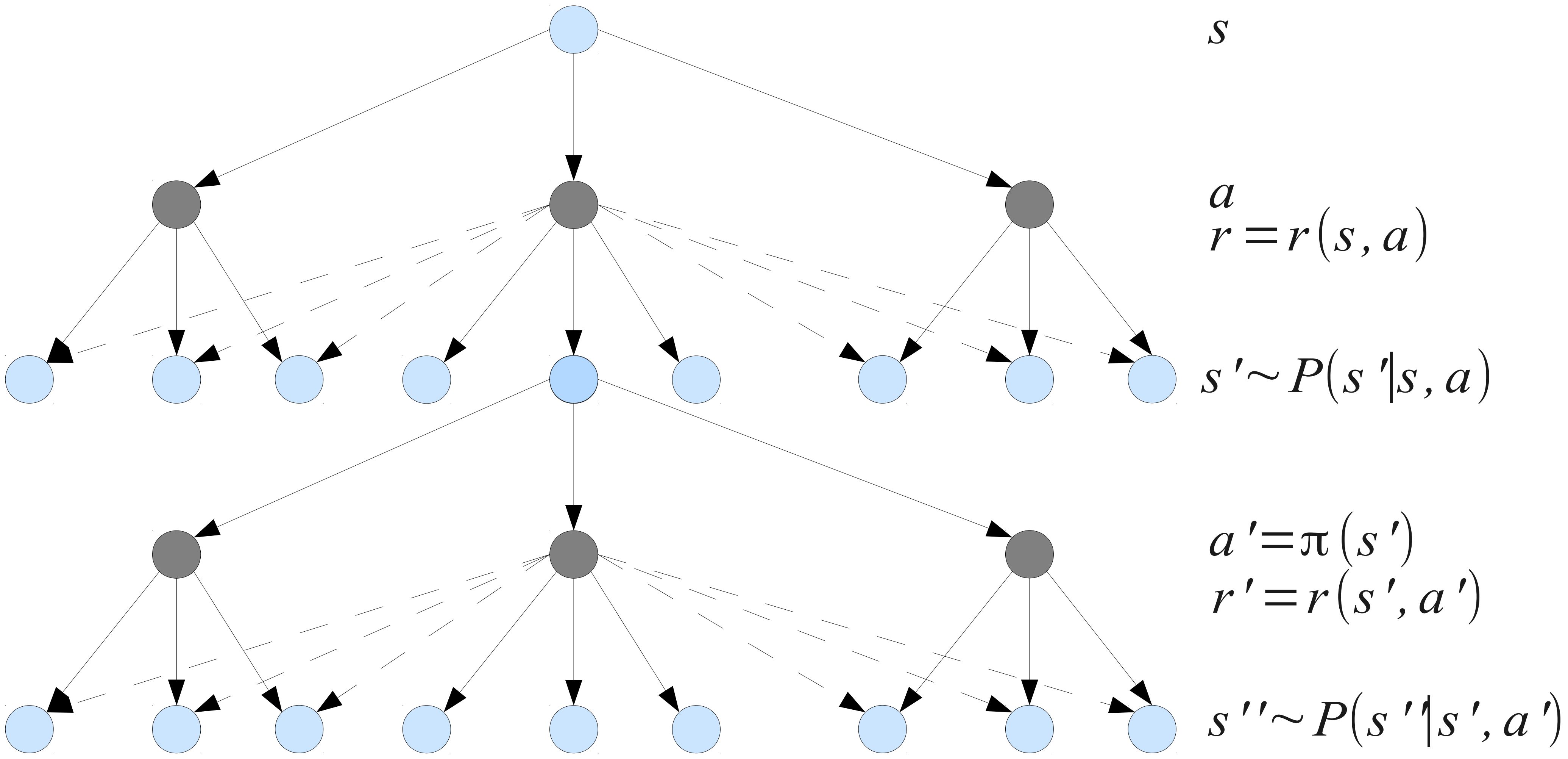
- where

$$Q^{\pi}(s,a) = r(s,a) + \sum_{s'} P(s'|s,a)\, Q^{\pi}(s', \pi(s'))$$

if we knew this then Q could be computed analytically

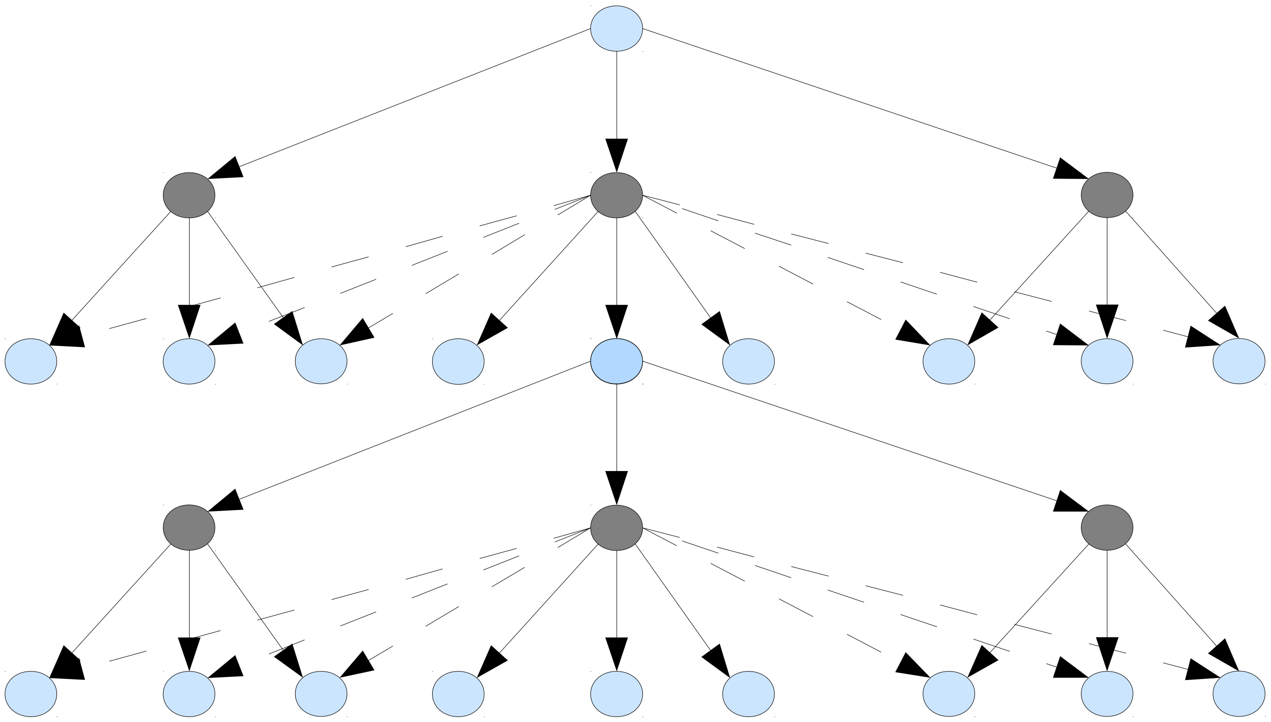
A tree representation of Q function

$$Q^\pi(s,a) = r(s,a) + \sum_{s'} P(s'|s,a)\, Q^\pi(s', \pi(s'))$$



Model based reinforcement learning

- Using dynamic programming can simplify the computation
- However, it is still to complex
- Therefore, some use only finite look ahead



Online reinforcement learning

- The model or the reward function is not always available

$$Q^{\pi}(s,a)=\cancel{r(s,a)}+\sum_{s'}\cancel{P(s'|s,a)}Q^{\pi}(s',\pi(s))$$

- However, we can observe a lot of dialogues

$$s_t\;,a_t\;,r_t$$
$$s_{t+1},a_{t+1},r_{t+1}$$
$$s_{t+2},a_{t+2},r_{t+2}$$
$$s_{t+3},a_{t+3},r_{t+3}$$
$$s_{t+4},a_{t+4},r_{t+4}$$

Unrolling Q(s,a)

Notice the dependence on t

- We want to get

$$Q^\pi(s_t,a_t)=E(r_t+Q^\pi(s_{t+1},a_{t+1}))$$

$$Q^\pi(s_t,a_t)=\sum_{s_{t+1}} P(s_{t+1}|s_t,a_t)(r_t+Q^\pi(s_{t+1},a_{t+1}))$$

$$Q^\pi(s_t,a_t)=\sum_{s_{t+1}} P(s_{t+1}|s_t,a_t)\,r_t+\sum_{s_{t+1}} P(s_{t+1}|s_t,a_t)Q^\pi(s_{t+1},a_{t+1})$$

$$Q^\pi(s_t,a_t)=r_t\sum_{s_{t+1}} P(s_{t+1}|s_t,a_t)+\sum_{s_{t+1}} P(s_{t+1}|s_t,a_t)Q^\pi(s_{t+1},a_{t+1})$$

$$Q^\pi(s_t,a_t)=r_t+\sum_{s_{t+1}} P(s_{t+1}|s_t,a_t)\,Q^\pi(s_{t+1},a_{t+1})$$

- Unrolling for $s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, s_{t+2}, a_{t+2}, r_{t+2}, s_{t+3}, a_{t+3}, r_{t+3}$

$$Q^\pi(s_t,a_t)=r_t+Q^\pi(s_{t+1},a_{t+1})$$

$$Q^\pi(s_{t+1},a_{t+1})=r_{t+1}+Q^\pi(s_{t+2},a_{t+2})$$

$$Q^\pi(s_{t+2},a_{t+2})=r_{t+2}+Q^\pi(s_{t+3},a_{t+3})$$

$$Q^\pi(s_{t+3},a_{t+3})=r_{t+3}+Q^\pi(s_{t+4},a_{t+4})$$

...

Can be solved by using Least Squares

MDP reinforcement learning

- It was assumed that
  - $Q^{\pi}(s, a)$ can be tabulated
  - however, this is intractable for billions of states

- Q(s,a) is therefore often approximated
  - $Q^{\pi}(s, a; \theta) \approx \theta^{T} \cdot \phi(s, a)$
  - or using, for example, Gaussian processes

- This also helps with using continuous states

Q(s,a) approximation

- Since $Q^{\pi}(s,a;\theta) \approx \theta^T \cdot \phi(s,a)$

$$\begin{aligned}
\theta^T \cdot \phi(s_t, a_t) &= r_t + \theta^T \cdot \phi(s_{t+1}, a_{t+1}) \\
\theta^T \cdot \phi(s_{t+1}, a_{t+1}) &= r_{t+1} + \theta^T \cdot \phi(s_{t+2}, a_{t+2}) \\
\theta^T \cdot \phi(s_{t+2}, a_{t+2}) &= r_{t+2} + \theta^T \cdot \phi(s_{t+3}, a_{t+3}) \\
\theta^T \cdot \phi(s_{t+3}, a_{t+3}) &= r_{t+3} + \theta^T \cdot \phi(s_{t+4}, a_{t+4}) \\
&\dots
\end{aligned}$$

- Again, this can be solved by least squares method
- This is called LS-SARSA
  - least squares – state action reward state action

SARSA – online RL

- Follow currently optimal policy
  - take a random action with probability e (Exploration)
    - to explore new possible efficient ways to control dialogue
- Estimate θ using samples
  - LS-SARSA
- Repeat, gradually decreasing e

POMDP reinforcement learning

- So far we considered only dialogue state **s**
- However, we should work with b(s)
  - probability distribution over dialogue states
- Since b(s) can be represented as vector with continuous values
  - continuous MDP methods can be used

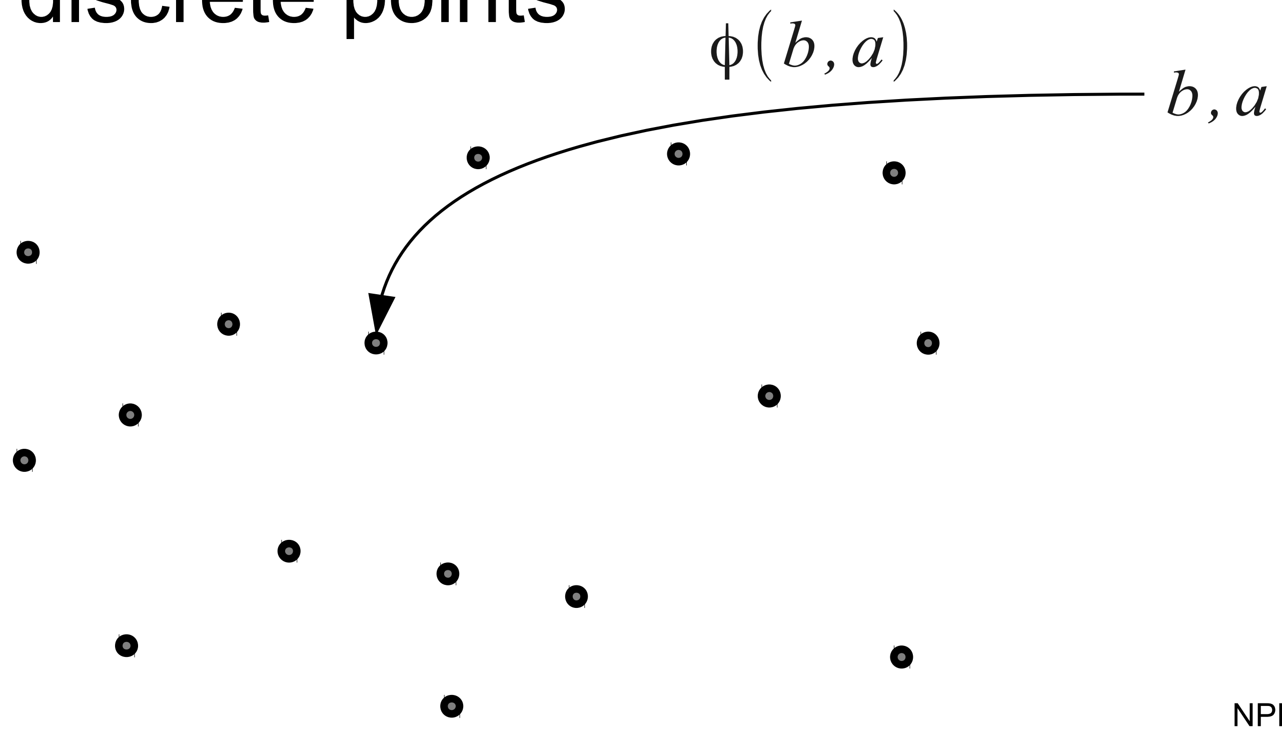
$$Q^{\pi}(b, a; \theta) \approx \theta^{T} \cdot \underbrace{\phi(b, a)}_{\text{Here it comes into play the feature extraction from the belief state}}$$

Grid based approach

- In the previous part of the lecture
  - Q function approximation was employed

$$Q^{\pi}(b,a;\theta) \approx \cancel{\theta^{T} \cdot \phi(b,a)}$$

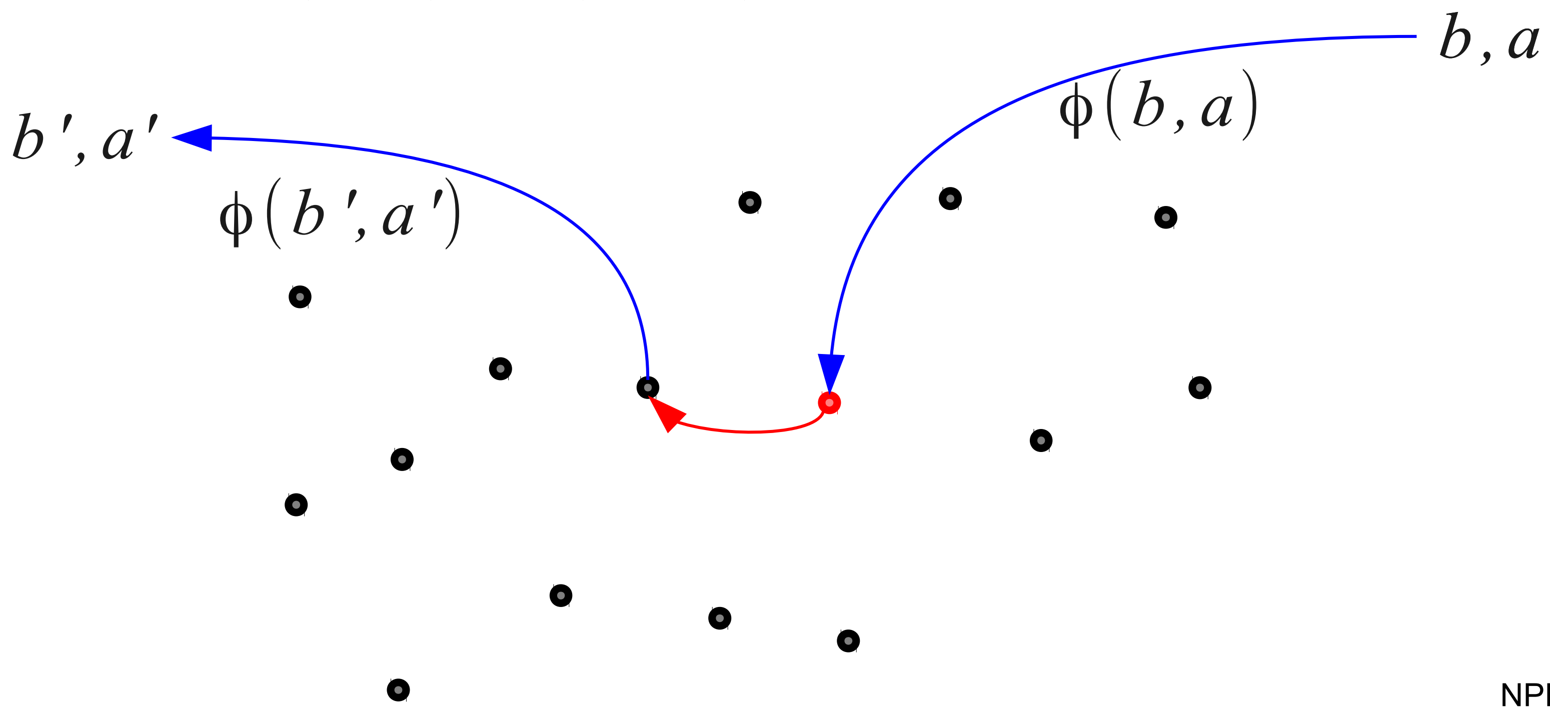
- In grid based approach, Q function is computed only at discrete points



Grid based approach

- Every time we need a Q value for $\phi(b, a)$
- We find a nearest point $\phi(b', a')$
  - based on some metric $m(\phi(b,a), \phi(b',a'))$
  - and use a Q value at the that nearest point

$$Q(b,a) \approx Q(b',a')$$



Learning

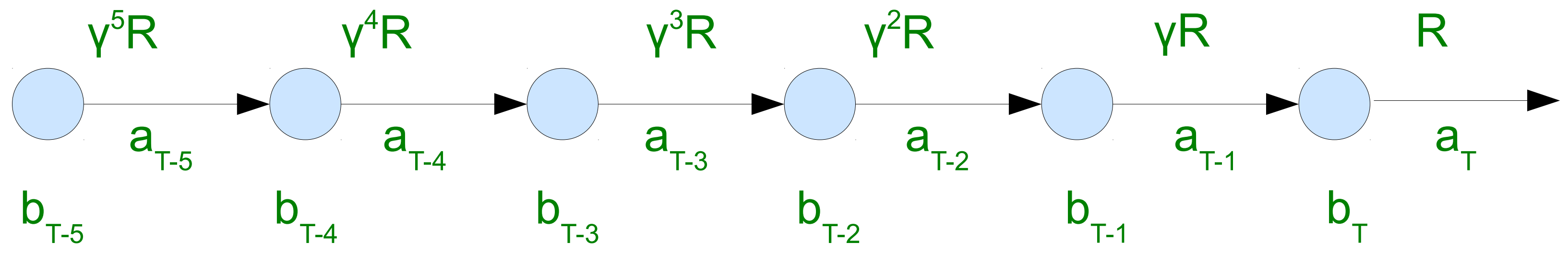
- Generate a dialogue using the current policy
  - collect **(b, a, r)** at each turn **t**

- Generate missing grid points, e.g.
  - if there is no grid point (b',a')
  - such that $m(\phi(b,a),\phi(b',a'))<\sigma$
  - then create a new grid point (b,a)

- The Q values are updated from the end of dialogue
  - for t from T until 1
    - find a nearest **(b',t')** for **(b,a)** at **t**
    - **Q(b',t') = (Q(b',t')*N(b',t') +R)/N(b',t')**
    - **N(b',t') += 1**
    - **$R \leftarrow \gamma R$**

Learning

- The main idea is to propagate the final reward toward all (b,a) visited in the dialogue



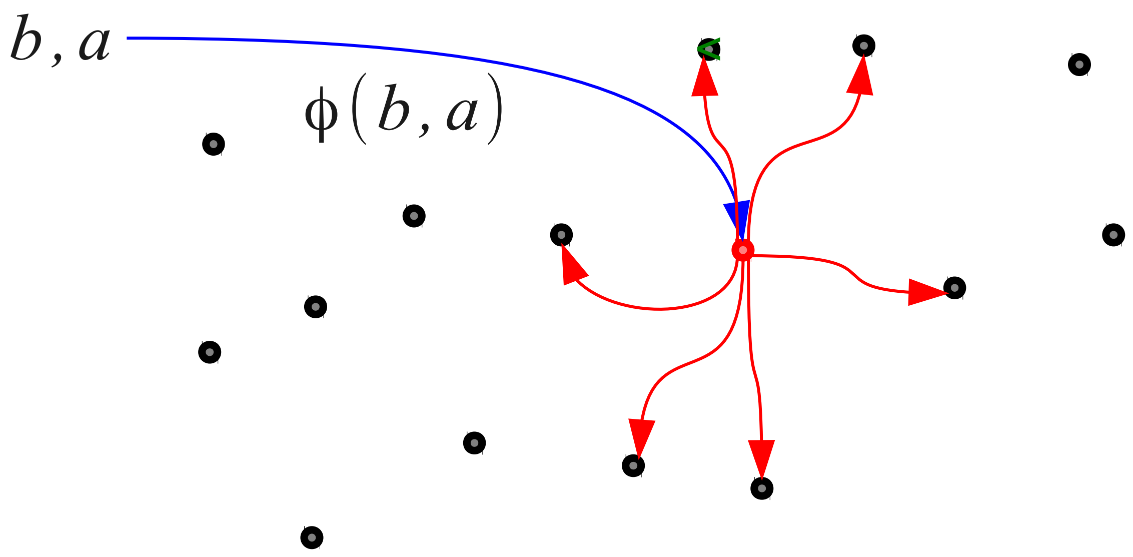
- And then use iterative update for an average

- The reward is discounted since the importance of the reward decreases with the distance

K-Means approach

- Although the previous approach looks a bit adhoc
  - it works
- However, more importantly it exhibits some interesting properties
  - it is non-parametric aka K-Means
- In K-Means approach, Q value does not depend only on one (b',a) but on K values which are then averaged

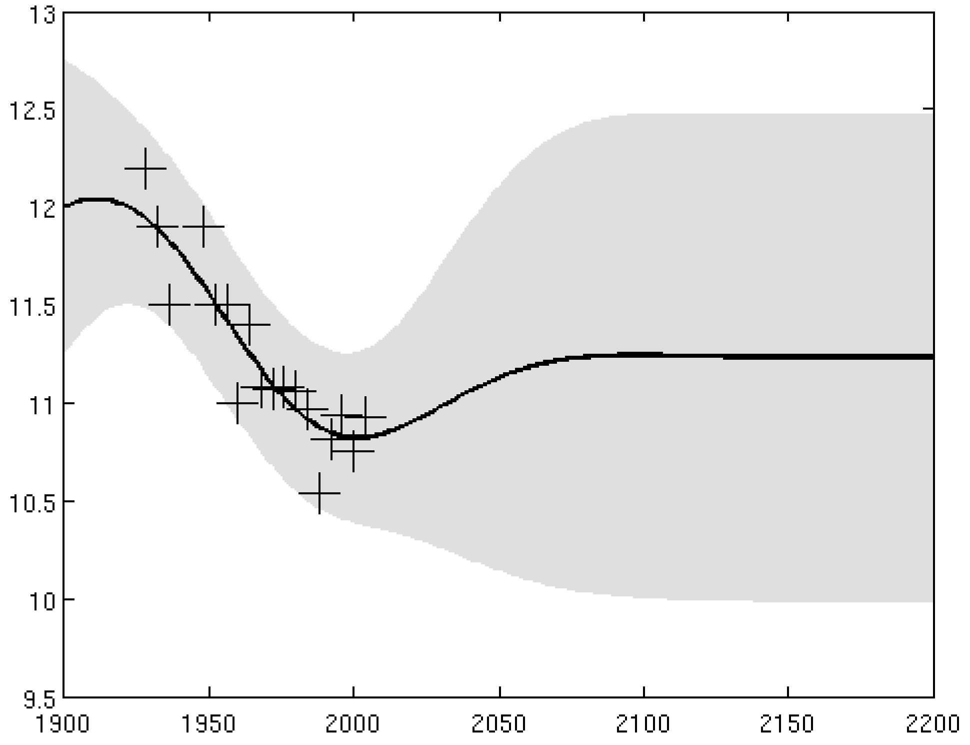
$$Q(b,a) \approx w_1 Q(b',a') + w_2 Q(b'',a'') + \ldots$$



Q function approximation using GP

- This can be generalised using Gaussian Processes

- Gaussian Processes are non-parametric Bayesian approach for regression



- The idea is that Q(b,a) depends on all observed rewards at all visited (b,a) points
  - and we are learning weights of theses grid points

Summary

- Generally, learning algorithms alternate between:
  - Estimating Value Function
  - Updating Policy
- Algorithms are typically
  - Model based
  - Online
- Due to complexity issues, approximations must be employed:
  - Summary features and actions
  - Function approximation

Thank you!

Filip Jurčíček

Institute of Formal and Applied Linguistics
Charles University in Prague
Czech Republic

Home page: http://ufal.mff.cuni.cz/~jurcicek