Probabilistic models of human-computer dialog

Jason D. Williams

Machine Intelligence Laboratory
Cambridge University Engineering Department

Goals of the talk

- **What is** a partially observable Markov Decision Process (POMDP)? **What's the difference** between POMDPs and other related machine learning techniques?
- **What's the relationship** between these techniques and various types of uncertainty?
- **Who** has applied POMDPs and related techniques to Spoken Dialog Systems and **how?**
- What are the **interesting research questions?**

(How am I attacking the problem?)

Agenda

- Part I: Algorithms & Methods
  - Belief networks, Influence diagrams
  - MDPs
  - Decision Theoretic methods
  - POMDPs
- Part II: Why use probabilistic methods for dialog management
  - Framing of the dialog management problem
  - Traditional approaches & new variants
  - The appeal of probabilistic methods
- Part III: Literature Search
  - MDPs
  - Decision Theoretic methods
  - POMDPs
- Part IV: My approach
  - A new type of data collection
  - Problem formulation
  - (Unsolved) example: Toy problem

Warnings!

- Lots of (virtual) hand-waving ….
  - Very little math discussed in most of this talk
- Focus on theoretical concerns …
  - May understate practical obstacles
- No references
  - Covered in detail in first-year report:

http://mi.eng.cam.ac.uk/~jdw30/willams2003PhDFirstYearReport.pdf

Part I

Some probabilistic techniques

Belief Networks – Preliminaries

- Any joint distribution can be factored

$$p(a,b,c,...) = p(a)\ p(b \mid a)\ p(c \mid a,b)\ ...$$

- Usually some/many dependencies can be removed
  - If *x* depends only on y, then:

$$p(x \mid y,...) = p(x \mid y)$$

Key ideas:

- Combine these notions intelligently to reduce number of required parameters
- Show inter-relationships graphically

Belief Networks – Definition

"A depends on B" can be restated as "B has a casual effect on A"

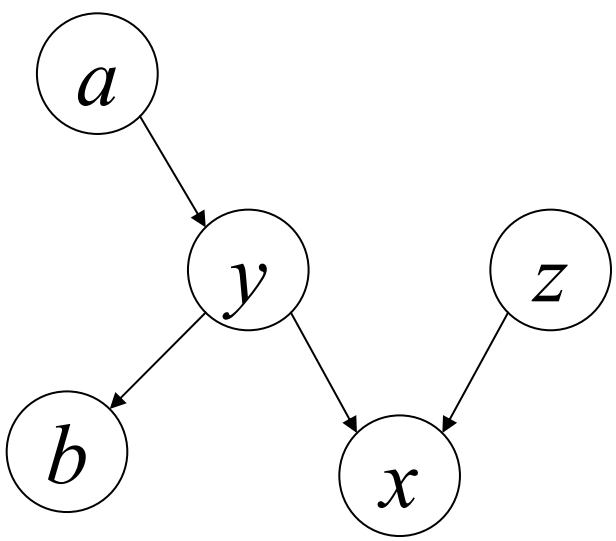
$p(x \mid y, z)$

$p(y \mid a)$

$p(b \mid y)$

$p(a)$

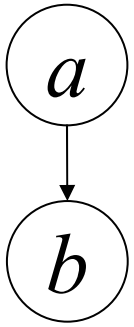
$p(z)$



- Definition: Bayesian Network:
  - Set of variables (i.e., nodes), each of which takes on a finite, mutually exclusive set of states
  - The nodes are connected by directed arcs into a connected DAG
  - Each node has a "potential table" (i.e., a conditional probability table)

Belief Networks – Evidence

a_1	0.1
a_2	0.9



	a_1	a_2
b_1	0.4	0.4
b_2	0.3	0.47
b_3	0.3	0.13

- Can rapidly calculate join probability table

$$p(A,B) = p(A)p(B \mid A)$$

	a_1	a_2
b_1	0.04	0.36
b_2	0.03	0.423
b_3	0.03	0.117

- Nodes have prior distributions…
  - "Best guess" in the absence of evidence

P(A) is explicit

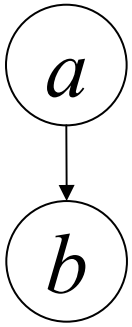
$$p(B) = \sum_A p(A,B)$$

$$p(B) = \sum_A p(A)p(B \mid A)$$

b_1	0.4
b_2	0.453
b_3	0.147

Belief Networks – Evidence

a_1	0.1
a_2	0.9



	a_1	a_2
b_1	0.4	0.4
b_2	0.3	0.47
b_3	0.3	0.13

- Nodes can receive "evidence"
  - "Hard" evidence

Suppose $B = b_3$; $p(A \mid B = b_3) = ?$

$$p(A \mid B) = \frac{p(A,B)}{p(B)}$$

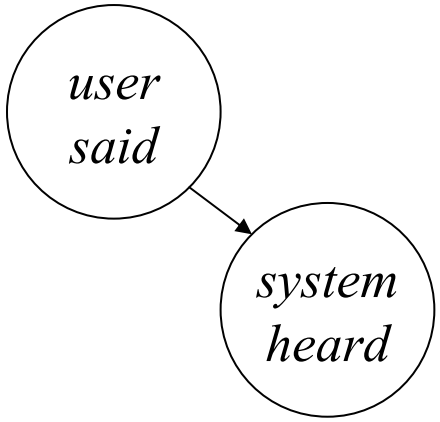
$$p(A \mid B) = \frac{p(A,B)}{\sum_{A'} p(A',B)}$$

$$p(A \mid B = b_3) = \frac{p(A,B = b_3)}{\sum_{A'} p(A',B = b_3)}$$

a_1	0.204
a_2	0.796

  - Can also have "Soft" evidence – a new distribution for A

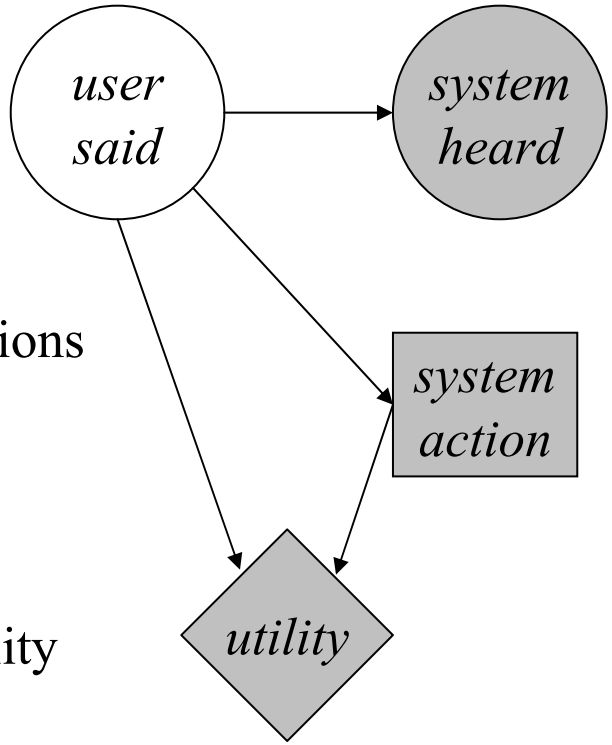
- There are a family of concepts describing how evidence is "transmitted" through the network.
- For us, we're usually concerned with working backwards to unseen, causal variables
- Example:

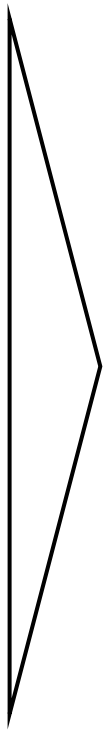


Influence Diagrams

An "influence diagram" extends a Bayesian Network to incorporate agent actions and consequences

- Hidden and observed variables
- Actions
- Utility



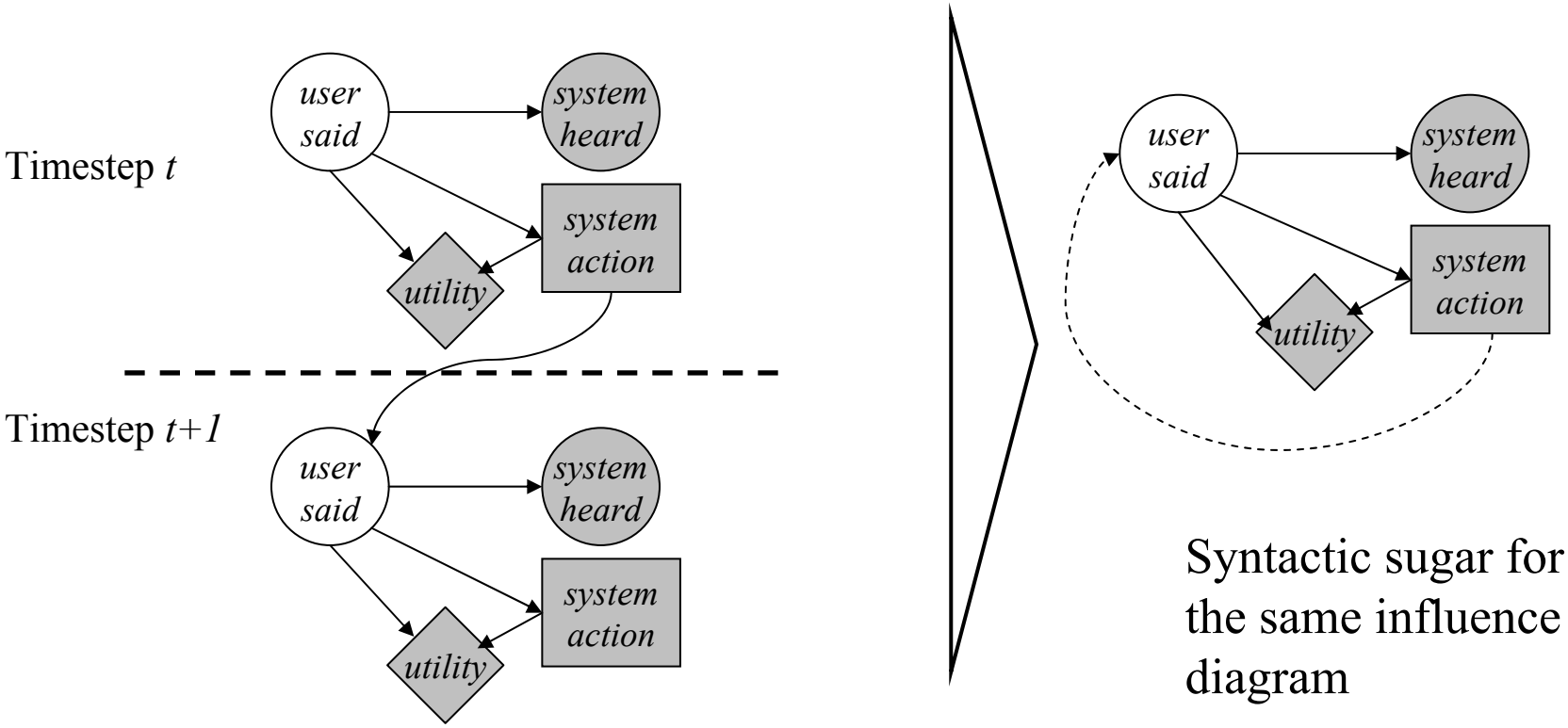


- Broad Goal: behavior under uncertainty by maximizing utility

Temporal Influence Diagrams

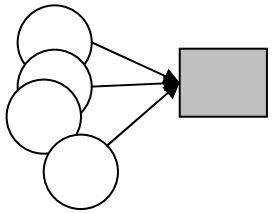
Incorporating history / Agents actions can effect the future

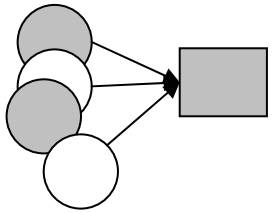
- The current state may depend on the past
  - Make the Markov assumption to limit history
- The agent's actions can have an effect on future states



Syntactic sugar for the same influence diagram

2 questions for models





- Is "state" fully observable, or partially observable?
  - "Fully observable" states can reflect degrees of certainty – but the "buckets" have to be derived somehow
  - Partially observable = "Belief State" – a continuous distribution over all states

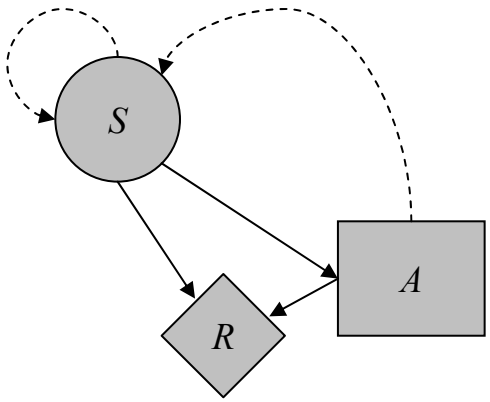
- Does the system plan?
  - A "greedy" immediate action selection will not seek a long-term goal
    - Long-term behavior needs to be "designed into" the rewards, state space, etc.
  - Cumulative reward – "Return" – forces the system to construct a plan – a rational – for choice of immediate actions

$$\max_A \{U_t\}$$

$$\max_A \{U_t + U_{t+1} + U_{t+2} + \ldots\}$$

(Fully Observable) Markov Decision Process

- States
- Actions
- Rewards



Non-trivial to learn or calculate the "policy" – mapping of state to action

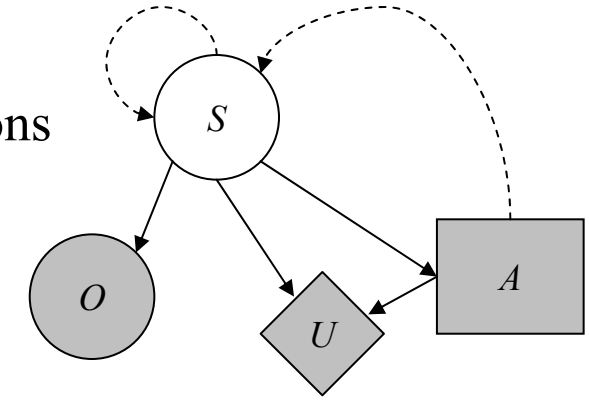
$$\pi(S) \Rightarrow A$$

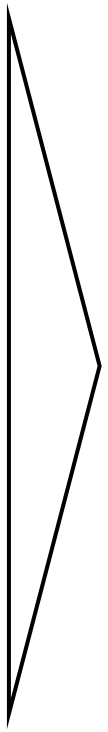
$$\pi^*(S) \Rightarrow A$$

- Goal is to maximize the "return"
- Note that state is "fully" (i.e., exactly) observed

Decision Theoretic Methods

- States
- Actions
- Rewards
- Observations



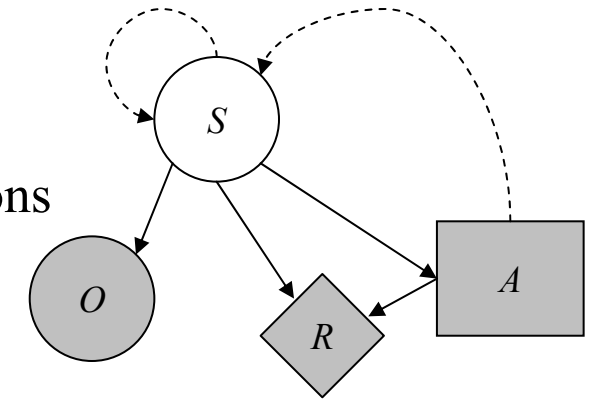


- Continuous state space
- Test each action
- Finding best action usually easy

- Goal is to maximize the immediate reward
- Note that state is "partially" observed
  - Note: I'm making assumptions about the structure of the state space to simplify the comparison

Partially Observable Markov Decision Process

- States
- Actions
- Rewards
- Observations



- Goal is to maximize the Return
- Note that state is "partially" observed
- Effectively a Continuous-space MDP

- State space is continuous
- A "policy" is a partitioning of |S| dimensional space
- Piecewise linear…
- … But number of segments proportional to depth of "look-ahead"
- Hard!

Asking our questions…

	MDP	DT	POMDP
Does "state" model uncertainty natively?	No	Yes	Yes
Does the system plan?	Yes	No	Yes

“Reinforcement learning”

• Model of agent/environment interaction • Two key properties:	(PO)MDP	DT
– “Goodness” is measured with numeric, evaluative feedback • Instructive feedback = supervised learning	Yes	Yes
– “Goodness” is measured over time with (possibly) delayed rewards	Yes	No

Reinforcement learning properties

- Designer states the problem and the desired goal(s)
  - Solution methods find (or approximate) optimal plans for any possible state (situation)
- Can learn "policy" on-line
  - System can improve automatically over time
    - (Provided it has constant access to reward function)
  - Policy can change as environment changes

Part II

Why use probabilistic techniques for SDSs?

Challenges for SDS

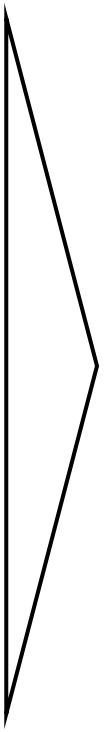
In a word: Uncertainty!

- *User-belief* uncertainty
  - What does the user want?
  - Not directly observable (by anyone)
- *ASR-channel* uncertainty
  - What did the user say?
  - Based on an observable value (i.e., reference transcription)
  - Channel errors: recognition, end-pointing
- *Dialog-state* uncertainty
  - What is the state of the conversation? What kind of contributions are sensible? What has been agreed on?
  - Not directly observable (by anyone)
  - Complicated by channel errors
- *Action-effect* uncertainty
  - What is the best action to take given the system's (long-term) goal?
  - Balancing between short-term and long-term needs: Planning

Traditional approaches

Some (familiar) examples

- *User-belief* uncertainty
  - Expert knowledge of task, Menus, Directed dialogs
- *ASR-channel* uncertainty
  - Confidence threshold, state-specific grammars
- *Dialog-state* uncertainty
  - Implicit/explicit confirmations, N-best lists
- *Action-effect* uncertainty
  - Expert knowledge of task, Dialog templates built by expert knowledge (VUI designers)



- Successful systems!
- Most approaches have focused on finding the right expressive constructs for dialog designers to quickly specify the dialog – e.g., VoiceXML, TRINDI, Mercury, RAVENCLAW

Newer approaches

Some applications of machine learning techniques

- *User-belief* uncertainty
  - Goal detection
    - Examples: (Gorin et al., 1997), (Chu-Carroll and Carpenter, 1999)
- *ASR-channel* uncertainty
  - Better confidence measures, incorporating many features
    - Examples: (Carpenter et al., 1999), (Krahmer et al., 1999 & 2001), (Hirschberg et al., 2001), (Litman et al., 2001), (Litman and Pan, 2000), (Langkilde et al., 1999)
- *Dialog-state* uncertainty
  - Dialog act detection
    - Examples: (Stolcke et al., 2000), (He and Young, 2003)
  - Ranked state hypotheses
    - Examples: (Lewin et al., 1998), (Higashinaka et al., 2003)
- *Action-effect* uncertainty
  - Selection of action based on a cost metric
    - Short vs. long term summarized nicely in (Dimiano and Traum, 2001)*
    - Example: (Dohaka, Yasuda and Aikawa, 2003)

* *Anticipatory planning for decision-theoretic grounding and task advancement in mixed-initiative dialogue systems*

The appeal of probabilistic techniques

Unify disparate probabilities into a single framework & solve with reinforcement learning

- *User-belief* uncertainty
- *ASR-channel* uncertainty
- *Dialog-state* uncertainty

Note that these are inter-related, and together naturally form a "state".

Probabilistic techniques allow us to combine these into a unified, factored distribution

- *Action-effect* uncertainty

Use reinforcement learning-based approaches to find best action to meet long-term goal

NB: I'm ignoring mathematical complexity at the moment!

Part III

Literature search

MDPs

- First presentations
  - (Levin and Pieraccini, 1997)
  - (Levin et al., 1998)
  - (Levin et al., 2000)
- Real systems
  - (Singh et al., 2002)
  - (Walker et al., 1998b)
- Other contributions
  - "Back-off" strategy for state-ting (Goddeau and Pineau, 2000)
  - User models for training (Pietquin and Renals, 2002), (Scheffler and Young, 2002)
  - More advanced training techniques (Scheffler and Young, 2002)
  - Bootstrapping with WoZ data (Williams and Young, 2003)

- Example formulation
  - For a vector of the following components:
    - Greet (y/n)
    - Active slot (0,1,2,3,4)
    - Confidence/confirmed (L, M, H confidence, Explicitly confirmed, Disconfirmed)
    - Value obtained for active slot (y/n)
    - Attempts on current slot (0,1,2)
    - Grammar (restrictive or non-restrictive)
    - Trouble-in-past (y/n)
  - Actions
    - System, Mixed, or User initiative prompt to ask/re-ask/confirm first 2 data fields
  - Reward function
    - Correct or incorrect submission of information (i.e., oracle)

Decision Theoretic Methods

- Key papers
  - (Paek and Horvitz, 2000a), (Horvitz and Paek, 2000), (Paek et al., 2000), (Paek and Horvitz, 2003)
- Example formulation from (Paek and Horvitz, 2000a)
  - Automated building receptionist
  - Focused on grounding problem
  - Network largely hand-crafted
  - Input nodes
    - Signal level
      - ASR confidence, threshold setting, overall parse fit, number of phrasal heads, number of non-terminals
    - Maintenance level
      - Whether telephone is in use, whether other people are in room, whether user's utterance includes name of system
    - Multi-modal
      - Whether the user is looking at the system (eye gaze), whether the user is moving toward or away from the system
  - Hidden nodes
    - Maintenance status
      - Values: No Channel, Channel but no signal, signal but no channel, channel and signal
    - Intention status
      - Binary value (Understanding semantic content of signals)
    - Conversation status
      - Binary value (Whether conversation as a whole is on track)
    - Grounding status (*)
      - Values: Ok, Channel failure, Signal failure, Intention failure, conversation failure
    - Activity goal (*)
      - Binary value (whether the user is participating in a joint activity with the system)
  - Utility
    - "Can be elicited from users through psychological experiments or direct assessment tools"
    - Based on action selected & hidden nodes marked (*)
  - Various grounding *strategies*
    - Strategy selected based on utility; Action (within strategy) selected based on Value of Information metric

Augmented MDPs

- (Roy et al., 2000): Augmented MDPs
  - Robot in a nursing home environment
  - Maintain belief state, but calculate a summary metric consisting of:
    - Single most likely state, and
    - Entropy over states
  - Thus a policy partitions only 1 real-numbered dimension
  - 13 states, 20 actions, 16 observations
  - Show that the Augmented MDP outperforms a standard MDP trained on the same data
    - Augmented MDP addresses uncertainty better

POMDPs with approximate solutions

- (Zhang et al., 2001): Approximate solutions
- Tour-guide system
- System attempts to discern & satisfy user's goal
  - 30 states
    - 6 user goals crossed with 5 "dialog" states
    - Dialog states include normal, silent, noisy, etc.
  - 18 actions
    - 6 domain actions
    - 12 dialog repair actions
  - Reward function for both dialog repairs & domain actions
  - 25 observations (the output of a Bayesian network)
    - No-signal-and-no-channel, no-signal, no-channel
    - Yes, no
    - 6 complete requests
    - 14 incomplete requests
- Assessment performed with user model
- Show that (simpler, cheaper) approximations which assume uncertainty will disappear in the future perform worse than (more expensive) approximations which don't

Literature summary

- (PO)MDPs and Decision Theoretic methods have been used to build real systems
- Clear support for techniques which explicitly model uncertainty at the dialog level
- Lots of interesting problems!

Part IV

My approach

Interesting questions

- What are the best hidden state elements?
- How can we collect data to build/train/bootstrap systems?
- How can we create a solid, encompassing theoretical framework which:
  - Allows incorporation of expert knowledge?
  - Avoids data scarcity?
  - Is tractable to train/solve?

My approach

1. Basics! Study the ASR channel in the abstract
   - WoZ experiments using a simulated ASR channel
   - Understand how people behave with an "ideal" dialog manager
     - For example, grounding model
     - Use these insights to inform state space and action set selection
   - Note that collected data has unique properties useful to:
     - RL-based systems
     - Hidden-state estimation
     - User modeling
2. Formulate dialog management problem as a POMDP
   - Decompose state into BN nodes – for example:
     - Conversation state (grounding state)
     - User action
     - User belief (goal)
   - Train using data collected
   - Solve using approximations

ASR channel vs. HH channel

Properties

HH dialog	ASR channel
• "Instant" communication • Effectively perfect recognition of words • Prosodic information carries additional information	• Turns explicitly segmented • Barge-in, End-pointed • Prosody virtually eliminated • ASR & parsing errors

Observations

• Frequent but brief overlaps • 80% of utterances contain fewer than 12 words; 50% < 5 • Approximately equal turn length • Approximately equal balance of initiative • About half of turns are ACK (often spliced)	• Few overlaps • Longer system turns; shorter user turns • Initiative more often with system • Virtually no turns are ACK • Virtually no splicing

Are models of HH dialog/grounding appropriate in the presence of the ASR channel?

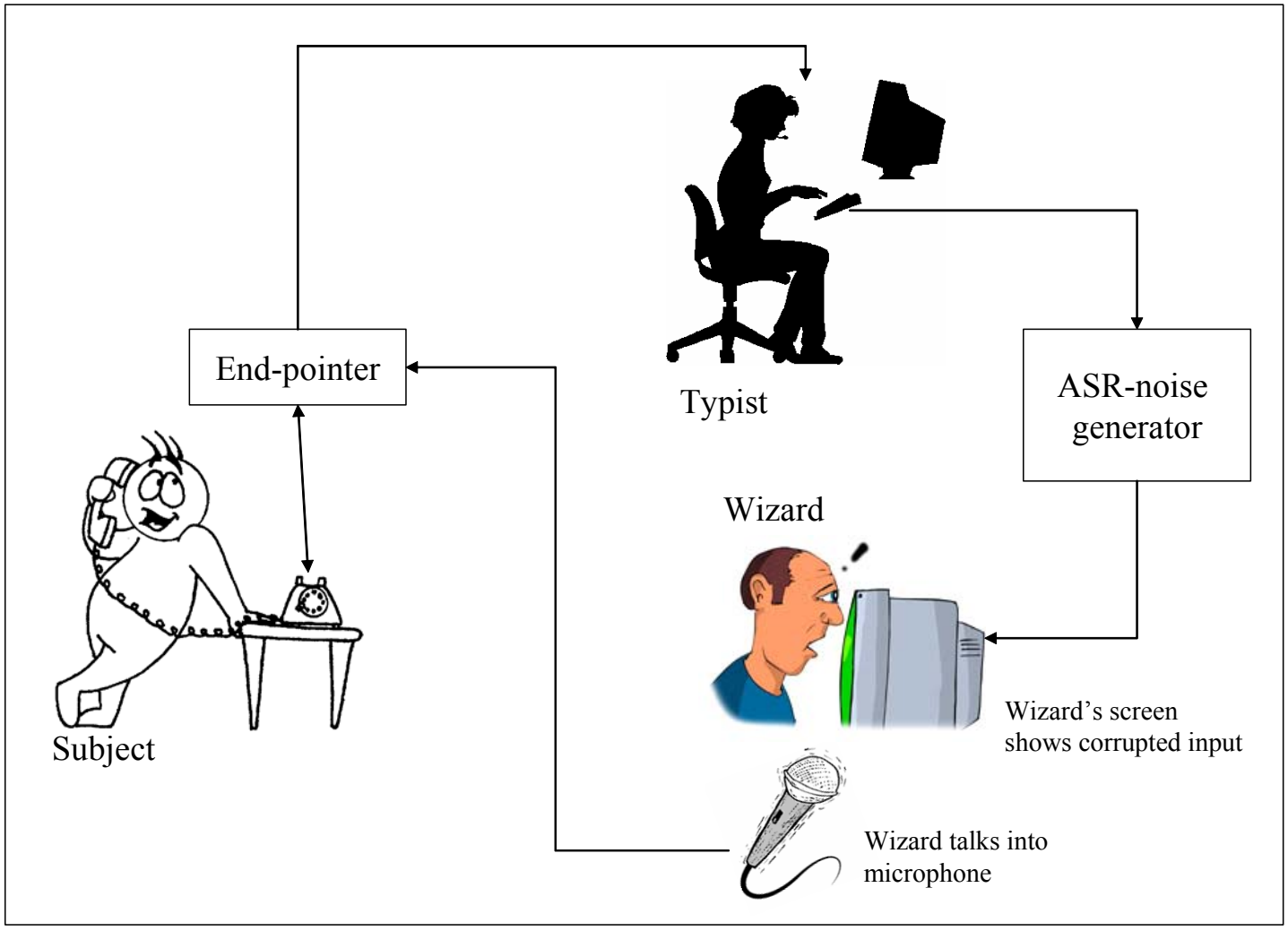
The paradox of "dialog data"

- To build a user model, we need to see the user's reaction to all kinds of misunderstandings
- However, most systems use a fixed policy
  - Systems typically do not take different actions in the same situation
  - Taking random actions is clearly not an option!
  - Constraining actions means building very complex systems…
- … and which actions should be in the system's repertoire?

An ideal data collection…

- …would show users reactions to a variety of error handling strategies (no fixed policy)
  - BUT would not be nonsense dialogs!
- …would use the ASR channel
- …would explore a variety of operating conditions – e.g., WER rate
- …would not assume a particular state space
- ... would somehow “discover” the set of system actions

Data collection set-up

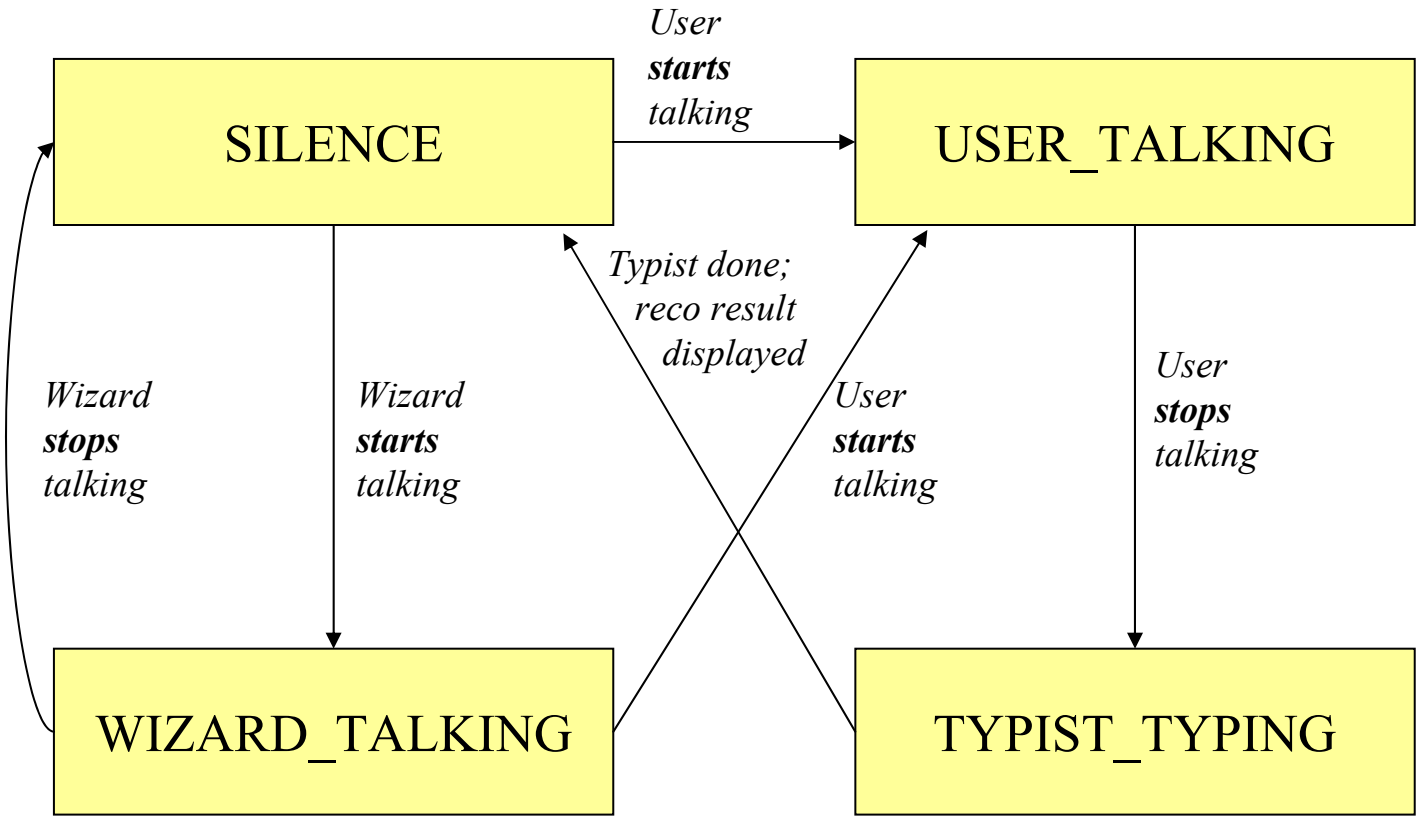


ASR simulation (1/2)

- Simplified FSM-based recognizer
- Flow
  - Reference input
  - Spell-checked against full dictionary
  - Converted to phonetic string using full dictionary
  - Phonetic lattice generated based on confusion model
  - Word lattice produced
  - Language model composed to re-score lattice
  - “De-coded” to produce word strings
  - N-Best list extracted.
- Various free variables to induce random behavior
- Seems to work reasonably, but hard to assess!

ASR simulation (2/2)

- Simple barge-in system
- User interrupts wizard



Scenario & Tasks

- Tourist / Tourist information scenario
  - Intentionally goal-directed
  - Intentionally simple tasks
  - Mixtures of simple information gathering and basic planning
- Wizard (Information giver)
  - Access to bus times, tram times, restaurants, hotels, bars, tourist attraction information, etc.
- User given series of tasks
- Likert scores asked at end of each task
- 4 Dialogs / user; 3 users/Wizard

Example task: Finding the perfect hotel

You're looking for a hotel for you and your travelling partner that meets a number of requirements. You'd like the following:

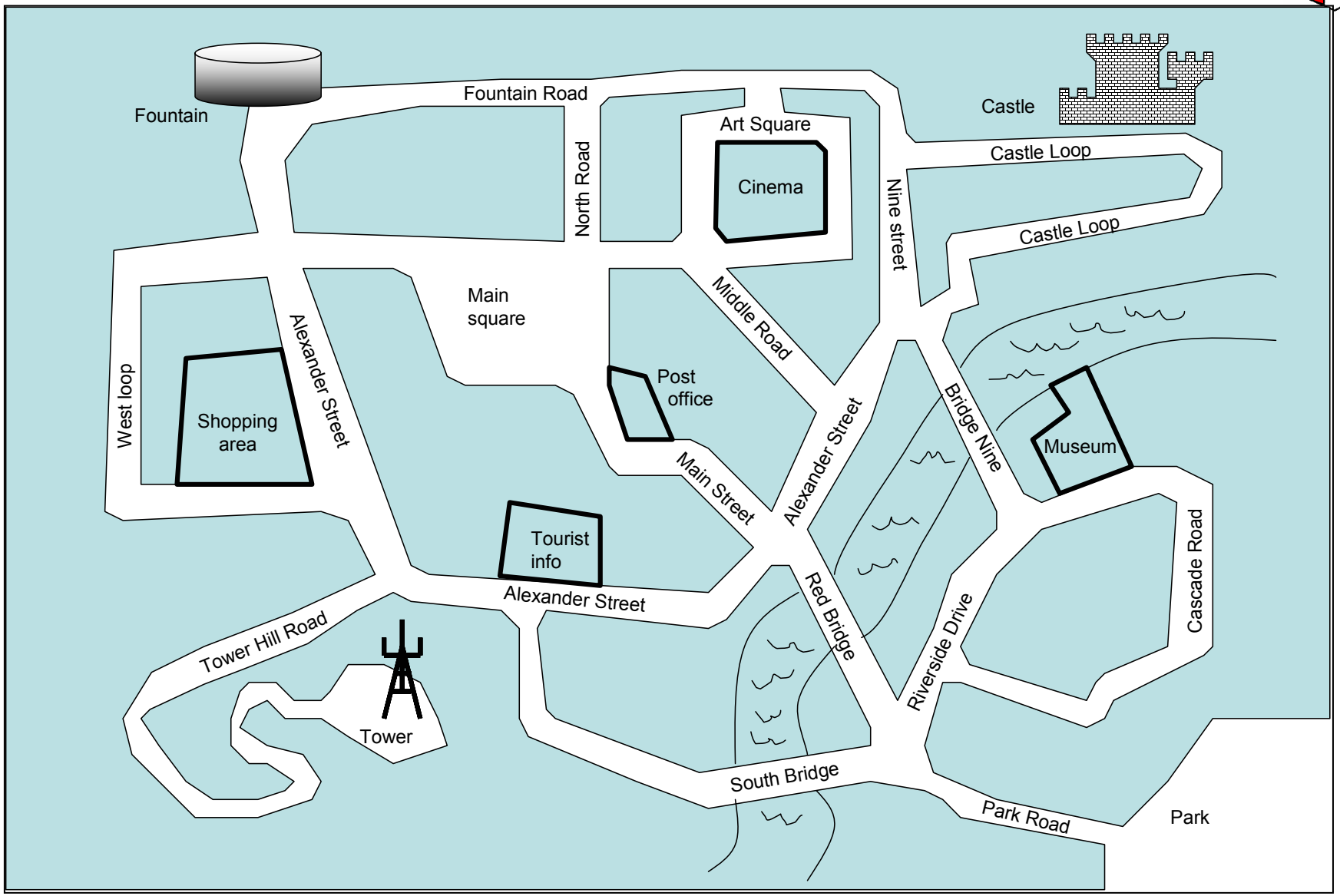
- En suite rooms
- Quiet rooms
- As close to the main square as possible

Given those desires, find the least expensive hotel. You'd prefer not compromise on your requirements, but of course you will if you must!

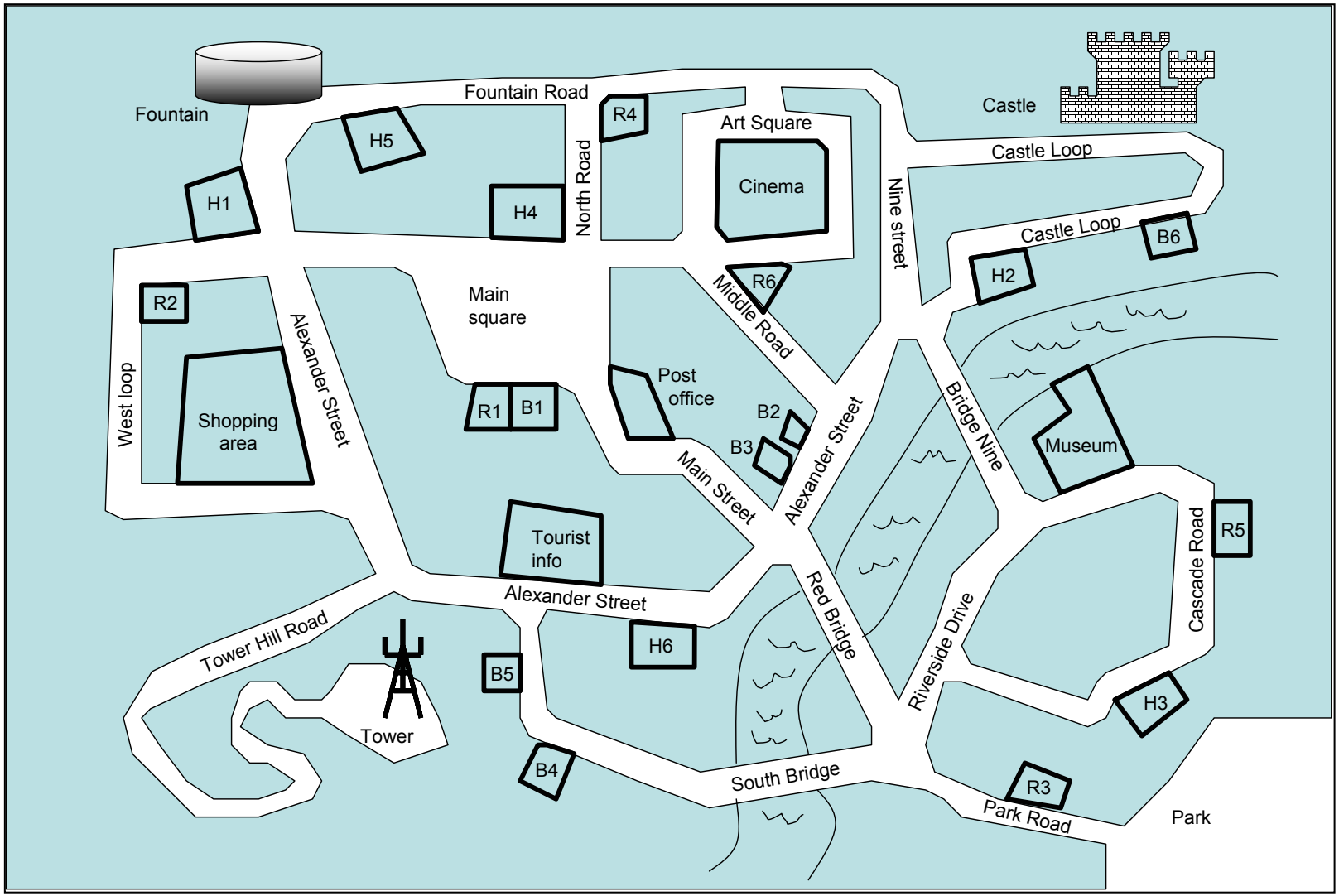
Please indicate the location of the hotel on the map and fill in the boxes below.

Name of accommodation	Cost per night for 2 people

User's Map


Fountain
Fountain Road
North Road
Art Square
Cinema
Castle
Castle Loop
Castle Loop
Nine street
Main square
Middle Road
West loop
Alexander Street
Shopping area
Post office
Main Street
Alexander Street
Bridge Nine
Museum
Tourist info
Alexander Street
Red Bridge
Riverside Drive
Cascade Road
Tower Hill Road
Tower
South Bridge
Park Road
Park

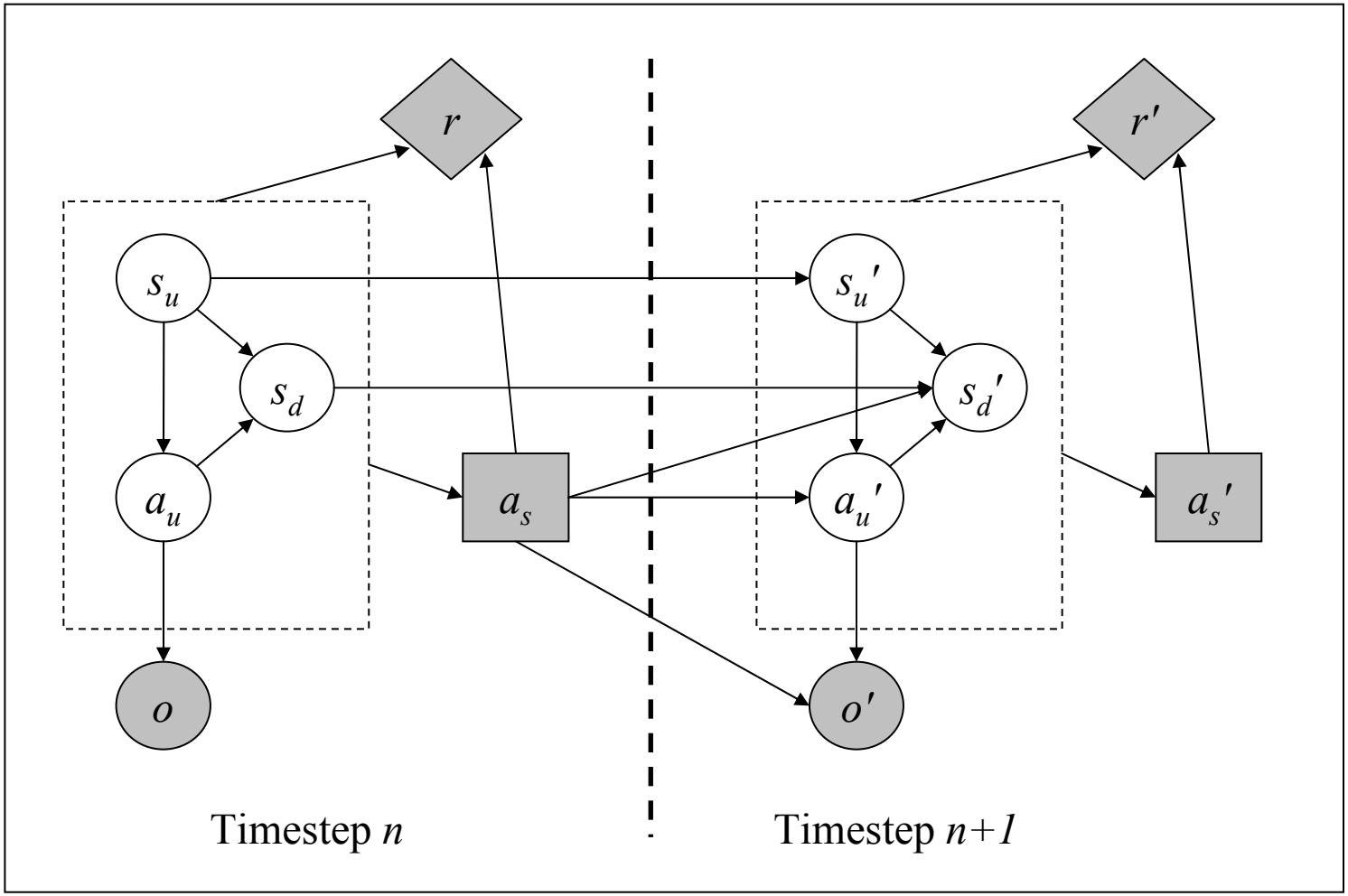
Wizard's map


Fountain
Fountain Road
H5
R4
Art Square
Castle
H1
H4
North Road
Cinema
Castle Loop
Nine street
Castle Loop
B6
H2
R6
R2
Main
square
Middle Road
Alexander Street
West loop
Shopping
area
Post
office
R1
B1
B2
B3
Main Street
Alexander Street
Bridge Nine
Museum
Tourist
info
Alexander Street
Red Bridge
Cascade Road
R5
Tower Hill Road
H6
Riverside Drive
B5
Tower
H3
B4
South Bridge
R3
Park Road
Park

Formulate dialog management as a POMDP

- Decompose state into BN nodes

Proposed dialog management Influence Diagram



Distributions required

- Values for nodes
  - Can be discovered from data collection
- Distributions
  - Can be estimated from data collected
    - Data sparsity?
  - Map to common-sense concepts:
    - User belief model
    - User action model
    - Dialog state model

Overview of the Toy problem

- There are three places called *a, b,* and *c*.
- The user's goal is to travel from one of these places to another – for example, if the user's goal is to travel from *a* to *b*, we write this as *(a,b)*.
- The machine's goal is to determine this pair as best it can through interacting with the user, and *submit* it – for example, to a ticket printing machine.

Toy Problem: (Machine) actions

- When interacting with the user, the machine can ***greet*** the user*, ***ask*** questions (e.g., "Where do you want to go from?") and make ***confirming*** statements (e.g., "So you want to go to ***a***, is that right?").
- At the end of the dialogue *(which is at a time of the machine's choosing),* the machine can either *submit* a pair of locations (e.g., ***submit-a-b)***, or ***fail***.

(*) The machine automatically greets the user during the first turn and after that, the "greet" action is no longer available.

Toy Problem: State space - overview

There are three components to the state space:

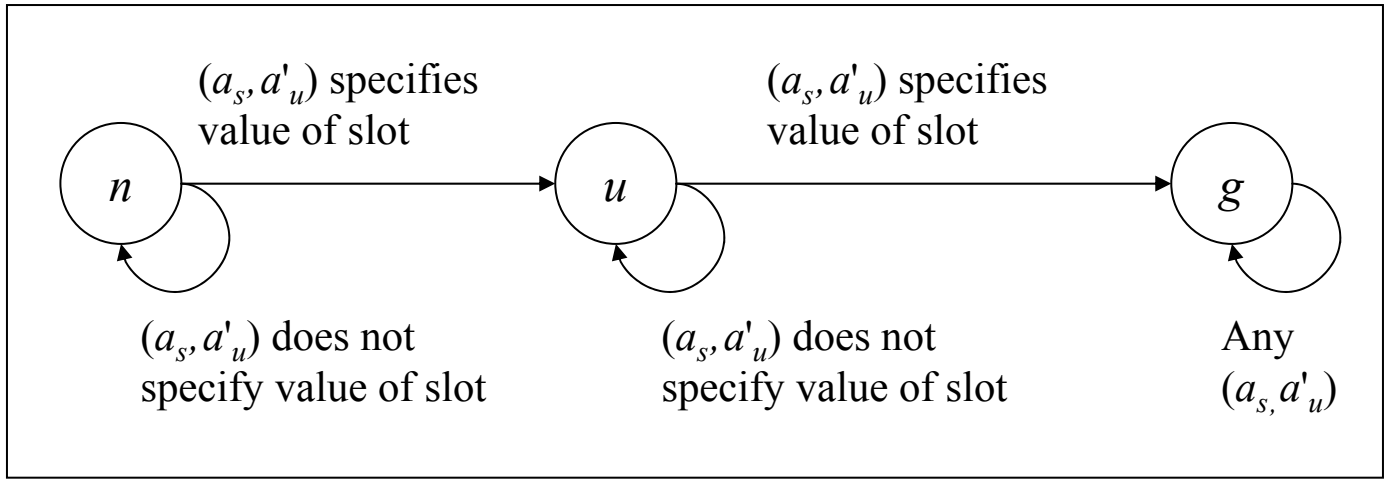
- The user's goal
- The user's action
- The state of the dialogue.

Toy Problem: State space – user's goal & action

- The user's goal is simply a pair of places, indicating *(from, to)*.
  - For example, *(a,b)*.
- The user's action may indicate:
  - Just a place (e.g., ***a***)
  - To or from a place, (e.g., ***from-a***)
  - To one place from another (e.g., ***from-a-to-b***)
  - ***Yes*** or ***No***
  - ***Null*** – which means the user didn't say or do anything

Toy Problem: State space – state of dialog

- We also have a state variable for the state of the *dialog*. This variable indicates whether the *from* and *to* components have been *grounded* or not. ***n*** means "not yet mentioned". ***u*** means "mentioned but not grounded". ***g*** means "grounded".
- Here we take "grounded" to mean "mentioned or confirmed by the user more than once."
- For example, if, so far in the conversation, *a* has been mentioned as the *from* location by the user, and *b* has been mentioned by the user *and confirmed* by the user, then the state of the dialog is ***(u,g)***.
- The diagram below shows a state transition diagram for the grounding model.



Toy Problem: Observations

- Observations are simply the user's actions combined with a measure of their recognition confidence, which may be ***h*** (high) *or* ***l*** (low).
- Example observations include:
  - **from-a-to-b-h**
  - **to-a-l**
  - **a-l**
  - **yes-l**
  - **no-h**
  - **null-h**

Toy Problem: Summary of variables & values

State variables & values

$$S = (s_u, s_d, a_u)$$

where:

$$s_d \in (d_1, d_2), d_i \in \{\text{n}, \text{u}, \text{g}\}$$

$$s_u \in (x_1, x_2), \quad x_i \in \{\text{a}, \text{b}, \text{c}\}, x_1 \neq x_2$$

$$a_u \in \begin{cases} x, \quad x \in \{\text{a}, \text{b}, \text{c}\} \\ \text{from–}x, \quad x \in \{\text{a}, \text{b}, \text{c}\} \\ \text{to–}x, \quad x \in \{\text{a}, \text{b}, \text{c}\} \\ \text{from–}x_1\text{–to–}x_2, \quad x_i \in \{\text{a}, \text{b}, \text{c}\}, x_1 \neq x_2 \\ \text{yes} \\ \text{no} \\ \text{null} \end{cases}$$

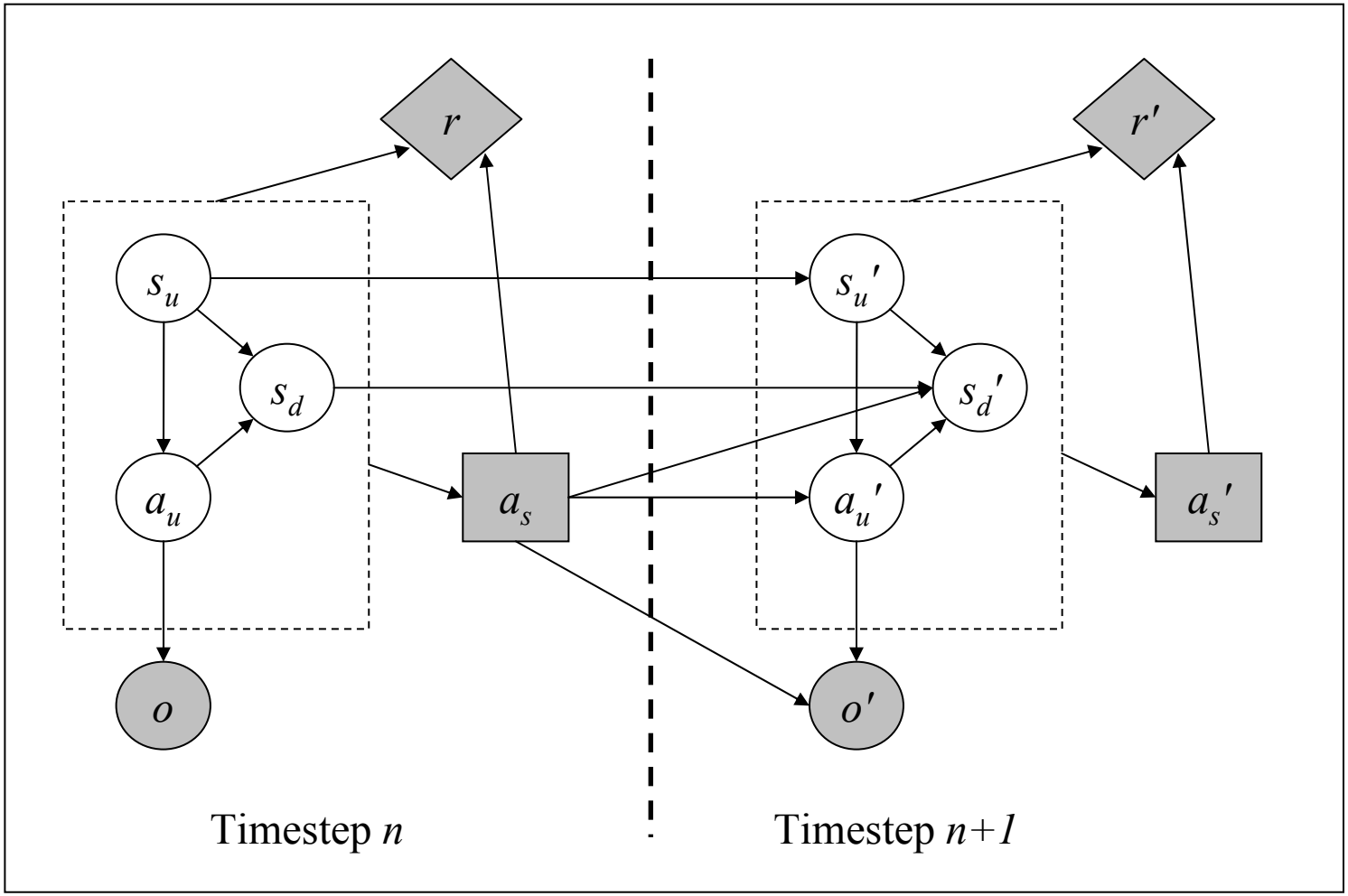
(System) action values

$$a_s \in \begin{cases} \text{greet} \\ \text{ask–to} \\ \text{ask–from} \\ \text{conf–from–}x, \quad x \in \{\text{a}, \text{b}, \text{c}\} \\ \text{conf–to–}x, \quad x \in \{\text{a}, \text{b}, \text{c}\} \\ \text{submit–}x_1\text{–}x_2, \quad x_i \in \{\text{a}, \text{b}, \text{c}\}, x_1 \neq x_2 \\ \text{fail} \end{cases}$$

Observation values

$$o \in \{x\text{–}y\}, \quad x \in a_s, y \in \{\text{h}, \text{l}\}$$

Influence diagram (recap)



Reward function - definition

The reward function is defined as:

$$r(S, a_s) = r(s_u, s_d, a_u, a_s) \Rightarrow \Re$$

The goal of the values of the reward function is both to incent the machine to achieve the goal as well as conform to conversational norms.

The system is rewarded for submitting the correct pair.

The system is penalized for submitting the wrong pair, and for taking the *fail* action.

In addition, the system is also penalized for asking for a field which has already been mentioned, confirming a field which has already been confirmed. There is also a small per-step penalty for any other action.

Reward function - values

$$r\left(s_u = (x, y), s_d, a_u, a_s = \text{submit}\text{–}x\text{–}y\right) = +10$$

$$r\left(s_u = (x, y), s_d, a_u, a_s \neq \text{submit}\text{–}x\text{–}y\right) = -10$$

$$r\left(s_u, s_d, a_u, a_s = \text{fail}\right) = -5$$

$$r\left(s_u, s_d = ([\text{u}, \text{c}], *), a_u, a_s = \text{ask}\text{–}\text{from}\right) = -3$$

$$r\left(s_u, s_d = (*, [\text{u}, \text{c}]), a_u, a_s = \text{ask}\text{–}\text{to}\right) = -3$$

$$r\left(s_u, s_d = (*, \text{c}), a_u, a_s = \text{conf-to-}x\right) = -2$$

$$r\left(s_u, s_d = (c, *), a_u, a_s = \text{conf-from-}x\right) = -2$$

And if none of the rules above apply, then:

$$r\left(s_u, s_d, a_u, a_s\right) = -1$$

Observation function – definition & values

The observation function is defined as:

$$p(o' \mid s', a) = p(o' \mid s'_u, s'_d, a'_u, a_s)$$
$$= p(o' \mid a'_u)$$
$$= p(o \mid a_u)$$

Where

$$p(o = (a\text{–l}) \mid a_u) = 0.35, \quad a = a_u$$
$$p(o = (a\text{–l} \mid a_u) \approx 0.015, \quad a \neq a_u$$
$$p(o = (a\text{–h}) \mid a_u) = 0.425, \quad a = a_u$$
$$p(o = (a\text{–h}) \mid a_u) \approx 0.004, \quad a \neq a_u$$

Transition function - definition

First we define three models:

- **The user belief model** $p(s'_u \mid s_u)$
- **The user action model** $p(a'_u \mid s'_u, a_s)$
- **The dialogue model** $p(s'_d \mid s_d, a_s, a'_u)$

The transition function is defined as:

$$p(s' \mid s, a) = p(s'_u, s'_d, a'_u \mid s_u, s_d, a_u, a_s)$$

Using the terms above, and conditional independence, we can re-write the transition function as:

$$= p(s'_u \mid s_u, s_d, a_u, a_s)\, p(a'_u \mid s'_u, s_u, s_d, a_u, a_s)\, p(s'_d \mid a'_u, s'_u, s_u, s_d, a_u, a_s)$$

$$= p(s'_u \mid s_u)\, p(a'_u \mid s'_u, a_s)\, p(s'_d \mid a'_u, s_d, a_s)$$

Transition function - values

The **user belief model** randomly assigns a (*from,to*) pair at the beginning of the task; after that, the user's goal does not vary

$$p(s'_u = s_u \mid s_u) = 1$$
$$p(s'_u \neq s_u \mid s_u) = 0$$

Transition function - values

The **dialogue model** is a deterministic probability distribution which gives the current state of a dialogue, given the user's action, the system's action, and the prior state of the dialogue

$$p(s'_d = (d_1, d_2) \mid a'_u = \text{null}, s_d = (d_1, d_2), a_s) = 1$$

$$p(s'_d = (\text{u}, d_2) \mid a'_u = x, s_d = (\text{n}, d_2), a_s = \text{ask–from}) = 1$$
$$p(s'_d = (\text{u}, d_2) \mid a'_u = \text{from–}x, s_d = (\text{n}, d_2), a_s) = 1$$
$$p(s'_d = (d_1, \text{u}) \mid a'_u = x, s_d = (d_1, \text{n}), a_s = \text{ask–to}) = 1$$
$$p(s'_d = (d_1, \text{u}) \mid a'_u = \text{to–}x, s_d = (d_1, \text{n}), a_s) = 1$$
$$p(s'_d = (\text{c}, d_2) \mid a'_u = x, s_d = (\text{u}, d_2), a_s = \text{ask–from}) = 1$$
$$p(s'_d = (\text{c}, d_2) \mid a'_u = \text{from–}x, s_d = (\text{u}, d_2), a_s) = 1$$
$$p(s'_d = (d_1, \text{c}) \mid a'_u = x, s_d = (d_1, \text{u}), a_s = \text{ask–to}) = 1$$
$$p(s'_d = (d_1, \text{c}) \mid a'_u = \text{to–}x, s_d = (d_1, \text{u}), a_s) = 1$$
$$p(s'_d = (\text{c}, d_2) \mid a'_u = x, s_d = (\text{c}, d_2), a_s = \text{ask–from}) = 1$$
$$p(s'_d = (\text{c}, d_2) \mid a'_u = \text{from–}x, s_d = (\text{c}, d_2), a_s) = 1$$
$$p(s'_d = (d_1, \text{c}) \mid a'_u = x, s_d = (d_1, \text{c}), a_s = \text{ask–to}) = 1$$
$$p(s'_d = (d_1, \text{c}) \mid a'_u = \text{to–}x, s_d = (d_1, \text{c}), a_s) = 1$$

$$p(s'_d = (d_1, \text{u}) \mid a'_u = \text{yes}, s_d = (d_1, \text{n}), a_s = \text{conf–from–}x) = 1$$
$$p(s'_d = (\text{u}, d_2) \mid a'_u = \text{yes}, s_d = (\text{n}, d_2), a_s = \text{conf–to–}x) = 1$$
$$p(s'_d = (d_1, \text{c}) \mid a'_u = \text{yes}, s_d = (d_1, \text{u}), a_s = \text{conf–from–}x) = 1$$
$$p(s'_d = (\text{c}, d_2) \mid a'_u = \text{yes}, s_d = (\text{u}, d_2), a_s = \text{conf–to–}x) = 1$$

$$p(s'_d = (d_1, d_2) \mid a'_u = \text{no}, s_d = (d_1, d_2), a_s = \text{conf–from–}x) = 1$$
$$p(s'_d = (d_1, d_2) \mid a'_u = \text{no}, s_d = (d_1, d_2), a_s = \text{conf–to–}x) = 1$$

$$p(s'_d = (\text{u}, \text{u}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{n}, \text{n}), a_s) = 1$$
$$p(s'_d = (\text{c}, \text{u}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{u}, \text{n}), a_s) = 1$$
$$p(s'_d = (\text{c}, \text{u}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{c}, \text{n}), a_s) = 1$$
$$p(s'_d = (\text{u}, \text{c}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{n}, \text{u}), a_s) = 1$$
$$p(s'_d = (\text{c}, \text{c}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{u}, \text{u}), a_s) = 1$$
$$p(s'_d = (\text{c}, \text{c}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{c}, \text{u}), a_s) = 1$$
$$p(s'_d = (\text{u}, \text{c}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{n}, \text{c}), a_s) = 1$$
$$p(s'_d = (\text{c}, \text{c}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{u}, \text{c}), a_s) = 1$$
$$p(s'_d = (\text{c}, \text{c}) \mid a'_u = \text{from–}x_1\text{–to–}x_2, s_d = (\text{c}, \text{c}), a_s) = 1$$

and if not specified above,

$$p(s'_d = (d_1, d_2) \mid a'_u, s_d = (d_1, d_2), a_s) = 1$$

where

$$x, x_1, x_2 \in \{a, b, c\}, x_1 \neq x_2$$
$$d_1, d_2 \in \{u, c, g\}$$

Transition function - values

The **user action model** specifies how a user responds to the system given the system's action and the user's goal.

$p(a'_u = \text{null} \mid s'_u = (x_1, x_2), a_s) = 0.1$

$p(a'_u = \text{from–}x_1\text{–to–}x_2 \mid s'_u = (x_1, x_2), a_s = \text{greet}) = 0.54$
$p(a'_u = \text{from–}x_1 \mid s'_u = (x_1, x_2), a_s = \text{greet}) = 0.18$
$p(a'_u = \text{to–}x_2 \mid s'_u = (x_1, x_2), a_s = \text{greet}) = 0.18$

$p(a'_u = \text{from–}x_1 \mid s'_u = (x_1, x_2), a_s = \text{ask-from}) = 0.225$
$p(a'_u = x_1 \mid s'_u = (x_1, x_2), a_s = \text{ask-from}) = 0.585$
$p(a'_u = \text{from–}x_1\text{–to–}x_2 \mid s'_u = (x_1, x_2), a_s = \text{ask-from}) = 0.09$

$p(a'_u = \text{to–}x_2 \mid s'_u = (x_1, x_2), a_s = \text{ask-to}) = 0.225$
$p(a'_u = x_2 \mid s'_u = (x_1, x_2), a_s = \text{ask-to}) = 0.585$
$p(a'_u = \text{from–}x_1\text{–to–}x_2 \mid s'_u = (x_1, x_2), a_s = \text{ask- to}) = 0.09$

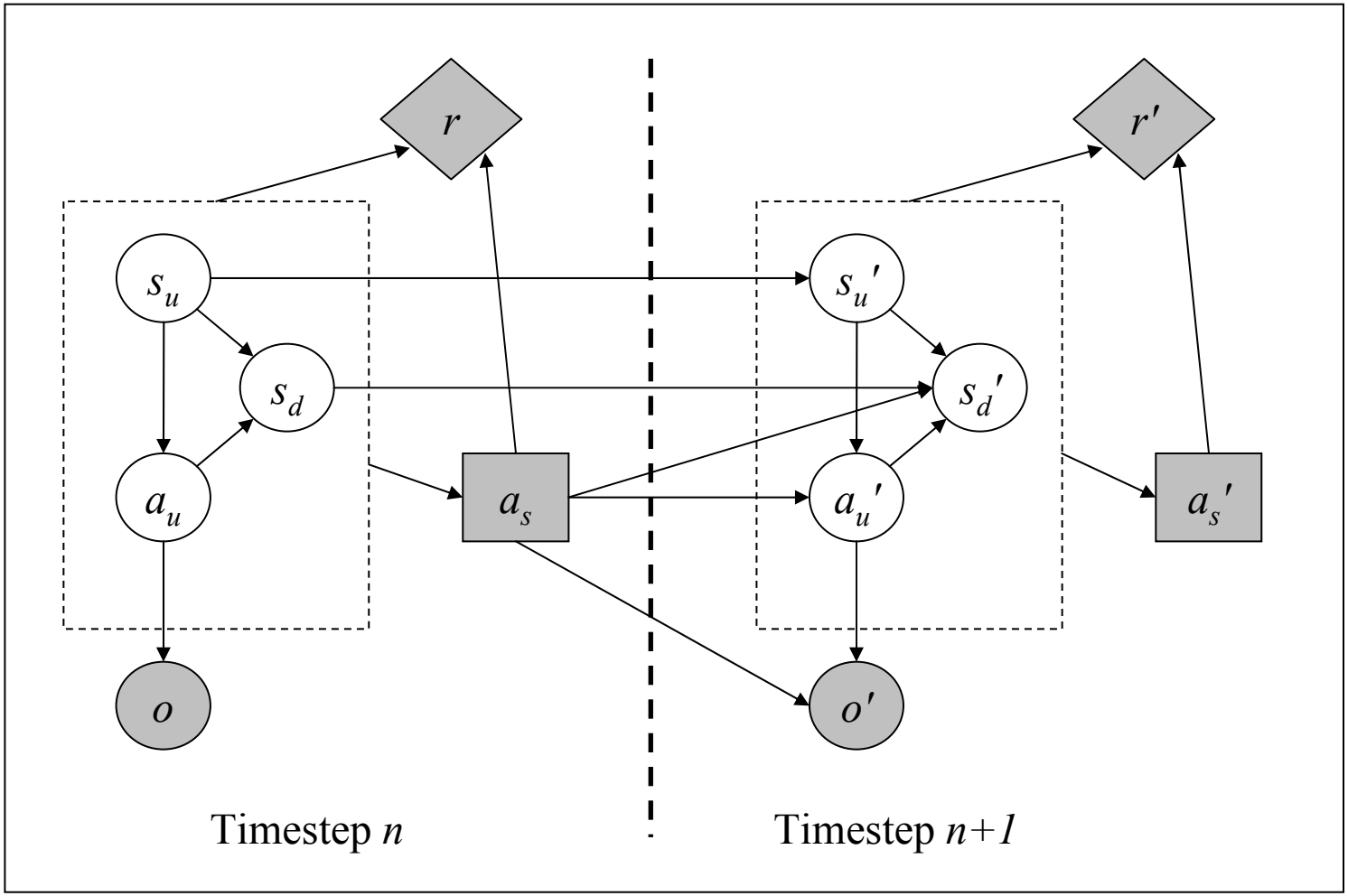
$p(a'_u = \text{from–}x_1 \mid s'_u = (x_1, x_2), a_s = \text{conf-from-}x_3) = 0.03375$
$p(a'_u = x_1 \mid s'_u = (x_1, x_2), a_s = \text{conf-from-}x_3) = 0.10125$
$p(a'_u = \text{yes} \mid s'_u = (x_1, x_2), a_s = \text{conf-from-}x_3) = 0.765, \quad x_1 = x_3$
$p(a'_u = \text{no} \mid s'_u = (x_1, x_2), a_s = \text{conf-from-}x_3) = 0.765, \quad x_1 \neq x_3$

$p(a'_u = \text{to–}x_2 \mid s'_u = (x_1, x_2), a_s = \text{conf-to-}x_3) = 0.03375$
$p(a'_u = x_2 \mid s'_u = (x_1, x_2), a_s = \text{conf-to-}x_3) = 0.10125$
$p(a'_u = \text{yes} \mid s'_u = (x_1, x_2), a_s = \text{conf-to -}x_3) = 0.765, \quad x_2 = x_3$
$p(a'_u = \text{no} \mid s'_u = (x_1, x_2), a_s = \text{conf-to-}x_3) = 0.765, \quad x_2 \neq x_3$

where

$x_1, x_2, x_3 \in \{a, b, c\}, x_1 \neq x_2$

Influence diagram: re-recap



Solutions…

- Exact solutions intractable
  - Witness algorithm runs for days
- Easy to compute belief state; hard to compute value function
  - Techniques to approximate value function most appropriate
  - ANN?

Thanks!

Jason D. Williams

jdw30@cam.ac.uk