Bayesian Networks: Representation and Application to Dialogue Systems

Some slides are adapted from Stuart Russell, Andrew Moore, or Dan Klein

www.monash.edu.au
Bayesian Reasoning: Learning Objectives

• Bayesian AI
• Bayesian networks
• Decision networks
Topics in Human Computer Interaction

Bayesian AI
Bayesian Decision Theory

- Frank Ramsey (1926)
- **Decision making under uncertainty** – what action to take when the state of the world is unknown
- **Bayesian answer** – find the utility of each possible outcome (action-state pair), and take the action that maximizes the *expected utility*
### Bayesian Decision Theory – Example

<table>
<thead>
<tr>
<th>Action</th>
<th>Rain (p=0.4)</th>
<th>Shine (1-p=0.6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take umbrella</td>
<td>30</td>
<td>10</td>
</tr>
<tr>
<td>Leave umbrella</td>
<td>-100</td>
<td>50</td>
</tr>
</tbody>
</table>

Expected utilities:
- \( E(\text{Take umbrella}) = 30 \times 0.4 + 10 \times 0.6 = 18 \)
- \( E(\text{Leave umbrella}) = -100 \times 0.4 + 50 \times 0.6 = -10 \)
Topics in Human Computer Interaction

Bayesian Networks
Bayesian Networks (BNs) – Overview

• **Introduction to BNs**
  – Nodes, structure and probabilities
  – Reasoning with BNs
  – Understanding BNs

• **Extensions of BNs**
  – Decision Networks
  – (Dynamic Bayesian Networks (DBNs))
Bayesian Networks: The Big Picture

• **Two problems with using full joint distribution tables as our probabilistic models:**
  – Unless there are only a few variables, the joint is too big to represent explicitly
  – Hard to learn (estimate) anything empirically about more than a few variables at a time

• **Bayes nets (aka graphical models): a technique for describing complex joint distributions (models) using simple, local distributions (conditional probabilities):**
  – Describe how variables interact locally
    > Local interactions chain together to give global, indirect interactions
Example Bayesian Network: Car
Graphical Model – Notation

- **Nodes**: variables (with domains)
  - Can be assigned (observed) or unassigned (unobserved)

- **Arcs**: interactions
  - Indicate “direct influence” between variables
  - Formally: encode conditional independence

- **For now, imagine that arrows mean direct causation**
Example: Coin Flips (I)

- N independent coin flips

\[ X_1 \quad X_2 \quad \ldots \quad X_n \]

- No interactions between variables: absolute independence
Example: Traffic (I)

- **Variables:**
  - R: It rains
  - T: There is traffic

- **Model 1:** independence

- **Model 2:** rain causes traffic

- **Why is model 2 better?**
Bayesian Networks – Definition (I)

- A data structure that represents the dependence between random variables
- A Bayesian Network is a directed acyclic graph (DAG) in which the following holds:
  1. A set of random variables makes up the nodes in the network
  2. A set of directed links connects pairs of nodes
  3. Each node has a probability distribution that quantifies the effects of its **parent nodes**
     > Discrete nodes have **Conditional Probability Tables (CPTs)**
- Gives a concise specification of the **joint probability distribution** of the variables
Bayesian Networks – Definition (II)

- The probability distribution for each node $X$ is a collection of distributions over $X$, one for each combination of its parents’ values:
  \[
  \text{Pr}(X|a_1, \ldots, a_n)
  \]
  - described by a *Conditional Probability Table (CPT)*
  - describes a “noisy” causal process

*Bayesian network = Topology (graph) + Local Conditional Probabilities*
Building the Joint Distribution

• We can take a Bayes’ net and build any entry from the full joint distribution it encodes by multiplying all the relevant conditional probabilities

\[ \Pr(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \Pr(x_i | \text{parents}(x_i)) \]

  – Typically, there’s no reason to build all the joint distribution – we build what we need on the fly

• Every BN over a domain implicitly defines a joint distribution over that domain, specified by local probabilities and graph structure

• But not every BN can represent every joint distribution
Building the Joint Distribution – Example

\[ \Pr(+\text{cavity}, +\text{catch}, \neg \text{toothache}) = ? \]

\[ = \Pr(\neg \text{toothache}|+\text{cavity}, +\text{catch}) \times \Pr(+\text{catch}|+\text{cavity}) \times \Pr(+\text{cavity}) \]

\[ = \Pr(\neg \text{toothache}|+\text{cavity}) \times \Pr(+\text{catch}|+\text{cavity}) \times \Pr(+\text{cavity}) \]
Example: Coin Flips (II)

Only distributions whose variables are independent can be represented by a Bayes net with no arcs.
Example: Traffic (II)

\[
\begin{align*}
\text{Pr}(+r, \neg t) &= \ ? \\
\text{Pr}(+r, \neg t) &= \text{Pr}(\neg t|+r) \text{ Pr}(+r) = \frac{1}{4} \times \frac{1}{4} = \frac{1}{16}
\end{align*}
\]
Example – Lung Cancer Diagnosis

A patient has been suffering from shortness of breath (called dyspnoea) and visits the doctor, worried that he has lung cancer.

The doctor knows that other diseases, such as tuberculosis and bronchitis are possible causes, as well as lung cancer. She also knows that other relevant information includes whether or not the patient is a smoker (increasing the chances of cancer and bronchitis) and what sort of air pollution he has been exposed to. A positive X-ray would indicate either TB or lung cancer.
Nodes and Values

Q: What do the nodes represent and what values can they take?

A: Nodes can be discrete or continuous

• Binary values
  – Boolean nodes (special case)
    Example: *Cancer* node represents proposition “the patient has cancer”

• Ordered values
  – Example: *Pollution* node with values *low, medium, high*

• Integral values
  – Example: *Age* with possible values 1-120
### Lung Cancer Example: Nodes and Values

<table>
<thead>
<tr>
<th>Node name</th>
<th>Type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pollution</td>
<td>Binary</td>
<td>{low, high}</td>
</tr>
<tr>
<td>Smoker</td>
<td>Boolean</td>
<td>{T, F}</td>
</tr>
<tr>
<td>Cancer</td>
<td>Boolean</td>
<td>{T, F}</td>
</tr>
<tr>
<td>Dyspnoea</td>
<td>Boolean</td>
<td>{T, F}</td>
</tr>
<tr>
<td>Xray</td>
<td>Binary</td>
<td>{pos, neg}</td>
</tr>
</tbody>
</table>
Lung Cancer Example: Network Structure

- Pollution
- Smoker
- Xray
- Dyspnoea

Cancer
Conditional Probability Tables (CPTs)

After specifying topology, we must specify the CPT for each discrete node

- Each row contains the conditional probability of each node value for each possible combination of values in its parent nodes
- Each row must sum to 1
- A CPT for a Boolean variable with n Boolean parents contains $2^{n+1}$ probabilities
- A node with no parents has one row (its prior probabilities)
Lung Cancer Example: CPTs
Reasoning with Numbers – Using Netica Software

![Diagram showing relationships between pollution, smoker status, cancer, X-ray results, and dyspnoea.](image)
Understanding Bayesian Networks

• **Understand how to construct a network**
  – A (more compact) representation of the joint probability distribution, which encodes a collection of conditional independence statements

• **Understand how to design inference procedures**
  – Encode a collection of conditional independence statements
  – Apply the *Markov property*
    > There are no direct dependencies in the system being modeled which are not already explicitly shown via arcs
    > Example: smoking can influence dyspnoea only through causing cancer
Representing Joint Probability Distribution: Example

\[
\Pr(P = \text{low} \land S = F \land C = T \land X = \text{pos} \land D = T) =
\]

\[
\Pr(P = \text{low}) \times \Pr(S = F) \times \Pr(C = T \mid P = \text{low}, S = F) \times \Pr(X = \text{pos} \mid C = T) \times \Pr(D = T \mid C = T)
\]
Reasoning with Bayesian Networks

- Basic task for any probabilistic inference system: Compute the posterior probability distribution for a set of *query variables*, given new information about some *evidence variables*
- Also called *conditioning* or *belief updating* or *inference*
Types of Reasoning

- **DIAGNOSTIC**
  - Query: P
  - Evidence: S, X, D

- **PREDICTIVE**
  - Evidence: P
  - Query: S, X, D

- **INTERCAUSAL** (explaining away)
  - Query: P
  - Evidence: S, X, D

- **COMBINED**
  - Evidence: P, S, X, D
Example – Earthquake (Pearl 1988)

You have a new burglar alarm installed. It reliably detects burglary, but also responds to minor earthquakes. Two neighbours, John and Mary, promise to call the police when they hear the alarm.

John always calls when he hears the alarm, but sometimes confuses the alarm with the phone ringing and calls then also. On the other hand, Mary likes loud music and sometimes doesn’t hear the alarm. Given evidence about who has and hasn’t called, you’d like to estimate the probability of a burglary.
# Earthquake Example: Nodes and Values

<table>
<thead>
<tr>
<th>Node name</th>
<th>Type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>B: Burglary</td>
<td>Boolean</td>
<td>{T,F}</td>
</tr>
<tr>
<td>A: Alarm (goes off)</td>
<td>Boolean</td>
<td>{T,F}</td>
</tr>
<tr>
<td>M: Mary calls</td>
<td>Boolean</td>
<td>{T,F}</td>
</tr>
<tr>
<td>J: John calls</td>
<td>Boolean</td>
<td>{T,F}</td>
</tr>
<tr>
<td>P: Phone rings</td>
<td>Boolean</td>
<td>{T,F}</td>
</tr>
<tr>
<td>E: Earthquake</td>
<td>Boolean</td>
<td>{T,F}</td>
</tr>
</tbody>
</table>


BN for Earthquake Example

- **Burglary**
  - $P(B=T) = 0.01$

- **Earthquake**
  - $P(E=T) = 0.02$

- **Phone Rings**
  - $P(P=T) = 0.05$

- **John Calls**

- **Mary Calls**

- **Alarm**

| B | E | $P(A=T|B,E)$ |
|---|---|-------------|
| T | T | 0.95 |
| T | F | 0.94 |
| F | T | 0.29 |
| F | F | 0.001 |

| A | $P(M=T|A)$ |
|---|-------------|
| T | 0.70 |
| F | 0.01 |
Causality?

• **When Bayesian networks reflect causal patterns:**
  – Often simpler (nodes have fewer parents)
  – Often easier to think about
  – Often easier to elicit from experts

• **BNs need not actually be causal, but it is good practice**
  – Arrows reflect correlation, not causation

• **What do the arrows really mean?**
  – Topology may happen to encode causal structure
  – Topology really encodes conditional independence
Example: Naïve Bayes

• Imagine we have one cause \( y \) and several effects \( x \):

\[
Pr(y, x_1, x_2, \ldots, x_n)
= Pr(y)Pr(x_1|y)Pr(x_2|y) \cdots Pr(x_n|y)
\]

• This is a naïve Bayes model
Size of a Bayes’ Net

• How big is a joint distribution over N Boolean variables? 
  \(2^N\)

• How big is an N-node net if each node has up to k parents? 
  \(O(N \times 2^{k+1})\)

• Both give the power to calculate \(Pr(X_1,X_2,\ldots,X_N)\), but BNs give huge space savings!

• Also easier to elicit local CPTs
Conditional Independence (reminder)

- X and Y are independent if
  \[ \forall x, y \Pr(x, y) = \Pr(x)\Pr(y) \]
  \[ \xrightarrow{- - - -} \quad X \perp Y \]

- X and Y are conditionally independent given Z
  \[ \forall x, y, z \Pr(x, y|z) = \Pr(x|z)\Pr(y|z) \]
  \[ \xrightarrow{- - - -} \quad X \perp Y|Z \]

- (Conditional) independence is a property of a distribution
Conditional Independence and BN Structure

• The relationship between conditional independence and BN structure is important for understanding how BNs work

• Factors that affect conditional independence
  + Causal chains
  + Common causes
  - Common effects
Causal Chains

- A causal chain of events

\[
Pr(x, y, z) = Pr(x) \cdot Pr(y|x) \cdot Pr(z|y)
\]

- Are X and Z independent given Y? **Yes!**

\[
Pr(z|x, y) = \frac{Pr(x, y, z)}{Pr(x, y)} = \frac{Pr(x) \cdot Pr(y|x) \cdot Pr(z|y)}{Pr(x) \cdot Pr(y|x)}
\]

\[
= Pr(z|y)
\]

- Evidence along the chain “blocks” the influence
Common Cause

- **Two effects of the same cause**
  - Y: Project due
  - X: Newsgroup busy
  - Z: Lab full

- **Are X and Z independent?** *No*

- **Are X and Z independent given Y?** *Yes!*

\[
\Pr(z|x, y) = \frac{\Pr(x, y, z)}{\Pr(x, y)} = \frac{\Pr(y) \Pr(x|y) \Pr(z|y)}{\Pr(y) \Pr(x|y)} = \Pr(z|y)
\]

- **Observing the cause blocks influence between the effects**
Common Effect

- **Two causes of one effect (v-structures)**
- **Are X and Z independent?**
  - Yes: the ballgame and the rain cause traffic, but they are not correlated
- **Are X and Z independent given Y?**
  - No: seeing traffic puts the rain and the ballgame in competition as explanation
- **This is backwards from the other cases**
  - Observing an effect activates the influence between possible causes

\[
\begin{array}{ccc}
X & \rightarrow & Y \\
\downarrow & & \downarrow \\
Z & \rightarrow & Y \\
\end{array}
\]

- X: Rain
- Z: Ballgame
- Y: Traffic
The General Case

• Any complex example can be analyzed using these three canonical cases
• General question: in a given BN, are two variables independent (given evidence)?
• Solution: analyze the graph
Direction-dependent Separation

- Graphical criterion of conditional independence
- We can determine whether a set of nodes \( X \) is independent of another set \( Y \), given a set of evidence nodes \( E \), via the Markov property
  - If a set of nodes \( X \) and a set of nodes \( Y \) are \textit{d-separated} by evidence \( E \), then \( X \) and \( Y \) are conditionally independent (given the Markov property)
- \textbf{D-separation is a property of the evidence}
**D-separation – Path**

- **Path (Undirected path):** A path between two sets of nodes $X$ and $Y$ is any sequence of nodes between a member of $X$ and a member of $Y$ such that every adjacent pair of nodes is connected by an arc (regardless of direction), and no node appears in the sequence twice.
D-separation – Blocked Path

• **Blocked path**: A path is *blocked* given a set of nodes $E$, if there is a node $Z$ on the path for which at least one of three conditions holds:

  1. $Z$ is in $E$ and $Z$ has one arrow on the path leading in and one arrow out (*chain*)
  2. $Z$ is in $E$ and $Z$ has both path arrows leading out (*common cause*)
  3. Neither $Z$ nor any descendant of $Z$ is in $E$, and both path arrows lead into $Z$ (*common effect*)

• A set of nodes $E$ *d-separates* two sets of nodes $X$ and $Y$, if every undirected path from a node in $X$ to a node in $Y$ is *blocked* given $E$
Determining D-separation

Chain

Common cause

Common effect
D-separation – Example (I)

\[ R \perp B \]
\[ R \perp B \mid T \]
\[ R \perp B \mid T' \]
D-separation – Example (II)

\[ L \perp T' | T \]

\[ L \perp B \]

\[ L \perp B | T \]

\[ L \perp B | T' \]

\[ L \perp B | T, R \]
D-separation – Example (III)

- **Variables:**
  - R: Raining
  - T: Traffic
  - D: Roof drips
  - S: I’m sad

- **Questions:**

  \[ T \perp D \]
  
  \[ T \perp D | R \]
  
  \[ T \perp D | R, S \]
Topics in Human Computer Interaction

Decision Networks
Decision Networks

- Extension of BNs to support making decisions
- Utility theory represents preferences between different outcomes of various plans
- Decision theory = Utility theory + Probability theory
Expected Utility

\[ EU(A \mid E) = \sum_{i} \Pr(O_i \mid E, A) U(O_i \mid A) \]

- \( E \) = available evidence
- \( A \) = a non-deterministic action
- \( O_i \) = a possible outcome state
- \( U \) = utility
A Decision network represents information about

- the agent’s current state
- its possible actions
- the state that will result from the agent’s action
- the utility of that state

Also called, *Influence Diagrams* (Howard & Matheson, 1981)
Types of Nodes

• **Chance nodes** – (ovals) random variables
  – Have an associated CPT
  – Parents can be decision nodes and other chance nodes

• **Decision nodes** – (rectangles) points where the decision maker has a choice of actions

• **Utility nodes (Value nodes)** – (diamonds) the agent's utility function
  – Have an associated table representing a multi-attribute utility function
  – Parents are variables describing the outcome states that directly affect utility
Types of Links

• **Informational Links** – indicate when a chance node needs to be observed before a decision is made

• **Conditioning links** – indicate the variables on which the probability assignment to a chance node will be conditioned
Fever Problem Description

Suppose that you know that a fever can be caused by the flu. You can use a thermometer, which is fairly reliable, to test whether or not you have a fever.

Suppose you also know that if you take aspirin it will almost certainly lower a fever to normal.

Some people (about 5% of the population) have a negative reaction to aspirin. You'll be happy to get rid of your fever, so long as you don't suffer an adverse reaction if you take aspirin.
Fever Decision Network
## Fever Decision Table

<table>
<thead>
<tr>
<th>Evidence</th>
<th>$Bel(FL=T)$</th>
<th>$EU(TA=yes)$</th>
<th>$EU(TA=no)$</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0.046</td>
<td>45.27</td>
<td>45.29</td>
<td>no</td>
</tr>
<tr>
<td>$Th=F$</td>
<td>0.018</td>
<td>45.40</td>
<td>48.41</td>
<td>no</td>
</tr>
<tr>
<td>$Th=T$</td>
<td>0.273</td>
<td>44.12</td>
<td>19.13</td>
<td>yes</td>
</tr>
<tr>
<td>$Th=T &amp; Re=T$</td>
<td>0.033</td>
<td>-30.32</td>
<td>0</td>
<td>no</td>
</tr>
</tbody>
</table>
Summary: Bayesian Networks

- Bayes nets compactly encode joint distributions
- **BNs are a natural way to represent conditional independence information**
  - **qualitative:** links between nodes – independencies of distributions can be deduced from a BN graph structure by D-separation
  - **quantitative:** conditional probability tables (CPTs)
- **BN inference**
  - computes the probability of query variables given evidence variables
  - is flexible – we can enter evidence about any node and update beliefs in other nodes
Reading and Software

**Reading**
  > Chapters 14.1-4, 16.1-2, 16.5
- *Bayesian Artificial Intelligence* (2nd ed), Chapman and Hall, Korb and Nicholson (2010),
  > Chapters 1, 2, 3.1-3.3 and 4.1-4.4

**Software**
Topics in Human Computer Interaction

Probabilistic and Statistical Dialogue Models

www.monash.edu.au
# Dialogue Management Approaches

<table>
<thead>
<tr>
<th>Formalism</th>
<th>Dialogue Grammars</th>
<th>Plan-based and Belief Desire Intention (BDI) models</th>
<th>Probabilistic and Statistical Models</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Grammars</td>
<td>Plans</td>
<td>Conditional probabilities</td>
</tr>
<tr>
<td>Processing method</td>
<td>Parser</td>
<td>Plan recognition</td>
<td>Statistical Inference</td>
</tr>
<tr>
<td>Performance</td>
<td>+ Efficient</td>
<td>+ Powerful</td>
<td>+ Efficient</td>
</tr>
<tr>
<td></td>
<td>– Restrictive</td>
<td>– Not so efficient</td>
<td>– Can go wrong</td>
</tr>
</tbody>
</table>

**Notes:**
- + Efficient
- – Restrictive
- + Powerful
- – Not so efficient
- – Can go wrong
Probabilistic and Statistical Dialogue Control

Approaches

• **Apply probabilistic principles to make decisions in dialogue**
  – Bayesian networks
  – Bayesian decision networks

• **Optimize and learn dialogue management strategies automatically using data from previous interactions and reward functions**
  – Markov Decision Processes (MDPs)
  – Partially Observable Markov Decision Processes (POMDPs)
Application of Bayesian Networks to Dialogue Systems
Bayesian Receptionist (Horvitz & Paek, 2000)

• Bayesian inference at different levels of an abstraction hierarchy to determine grounding
  – Conversation, Intention, Signal, Channel
• Decision-theoretic strategies to guide the conversation based on expected utility
  – treat grounding as decision making under uncertainty
• Value of Information (VoI) analysis to identify actions that maximize grounding
Levels of Representation

**Conversation Control**
- $S$ is proposing activity $\alpha$ to $L$
- $L$ is considering $S$'s proposal of $\alpha$

**Conversation Level**

**Intention Module**
- $S$ is signaling that $p$ for $L$
- $L$ is recognizing that $p$ from $S$

**Intention Level**

**Maintenance Module**
- $S$ is presenting signal $\sigma$ to $L$
- $L$ is identifying signal $\sigma$ from $S$

**Signal Level**
- $S$ is executing behavior $\beta$ for $L$
- $L$ is attending to behavior $\beta$ from $S$

**Channel Level**
Conversation Control

- Assesses the status of key variables in all of the modules
- Decides where to focus on resolving uncertainties, and what grounding actions to take in light of their likely costs and benefits
- Maintains a historical record of the dialog
  - including factors such as the number of repair actions taken in prior states of grounding
- Monitors its own performance and readjust its uncertainties and utilities
Maintenance Module BN
Conversational Module BN
Grounding Actions

- Do conditionally relevant action
  - No repair
    - Assume speech is overheard and ignore
    - Wait for further information before deciding
    - Give positive feedback (acknowledgment)
      - Display confusion (to elicit user-initiated repair)
        - In a general way
          - Specify grounding level
        - In a general way
          - Specify grounding level
      - Seek clarification
        - In a general way
          - Specify grounding level
      - Confirm understanding
        - Declare intention before action
    - Consider combinations of repair actions
  - Other Repairs...
Refinement through VoI

- **VoI analysis identifies the best evidence to observe in light of the inferred probabilities**

- **Computing VoI:**
  - For every observation, calculate the expected utility of the best decision associated with each value the observation can take considering the likelihood of different values

- **Once VoI recommends a piece of evidence to observe, the system can**
  - phrase a question to solicit that information or
  - make a recommendation
Example – Providing Services

I am here to visit Fred Smith way you contact in.

Conversation Control

User’s Goal

Intention Module

Maintenance Module

Visit1 (+Pres +TI +Weak Oblig +Nov +Meet +loc sr)

I\:I (+Pres1 +Sing +Anim +Hnum)

Deb\:eay1 (+Pers1 +Sing +Count +Mass +Obj)

Mode\:Fred Smith (+Hasn +Pers3 +Sing +PreqN +nO +Anim +Conc +Hnum +Hase +Count)

FIRSTNAME\:Fred1 (+Hasn +Pers3 +Sing +PreqN +nO +Anim +Conc +Fname +Hnum +Hase)

LASTNAME\:Smith1 (+Pers3 +Sing +PreqH +Anim +Conc +Hname +Hase +Count)

Fact\:Evry\:personal

Att\:contact1 (+Pres +Blocked +IO + loc sr)

Sub\:you1 (+Pers2 +Plur +Anim +Hnum)

In\:eay1

Mode\:here1 (+IO +Loc)

I will call Fred Smith for you right away.
Topics in Human Computer Interaction

Markov Decision Processes
Dialogue as a Markov Decision Process

\((S, A_s, T, R, \gamma, \pi)\)

- **S** – a set of states describing the world of the agent
- **\(A_s\)** – a set of actions an agent may take
- **\(T\)** – a set of transition probabilities \(\Pr_{T}(s_t | s_{t-1}, a_{t-1})\)
  - A dialogue is a sequence of state-action pairs \((s_0, a_0)\) \((s_1, a_1)\) \((s_2, a_2)\) ...... \((s_n, a_n)\)
  - Assumption: state transitions are Markovian
    \[
    \Pr(s_{t+1}|s_t, s_{t-1}, ..., s_0, a_t, a_{t-1}, ..., a_0) = \Pr(s_{t+1}|s_t, a_t)
    \]
- **\(R\)** – an immediate reward \(r(s, a)\) associated with action \(a\) in state \(s\)
- **\(\gamma\)** – a discount factor \(0 \leq \gamma \leq 1\)
- **\(\pi\)** – the policy, a mapping between the state space and the action set → Find the optimal policy \(\pi^*\)
Utility Function

• Expected cumulative reward $U(s)$ for state $s$ can be computed by the *Bellman equation*:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} \{ \Pr(s'|s, a) U(s') \}$$

immediate reward in state $s$

expected discounted utility of all possible next states $s'$, assuming that once there we take the optimal action

• **Discount makes the agent care more about current than future rewards**
  – the farther in the future is a reward, the more discounted is its value
Value Iteration Algorithm

- **Input:** MDP($S, A_s, T, R, \gamma$)
- **Variables:** Utility vectors $U$, initially 0
- **Returns:** a utility function

1. Repeat until maximum utility change < Thrsh
   
   $U \leftarrow U'$
   
   For each state $s$ in $S$ do
   
   $U'(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} \Pr(s'|s, a) U(s')$

2. return $U$
Why Reinforcement Learning?

- Sometimes the state space is very large (or infinite)
- The MDP parameters may not be known in advance:
  - Transition model and reward function are typically unknown

→ We need to resort to learning
Reinforcement Learning

- Systematic exploration of all the different actions a system can take
- The system learns by being rewarded for making choices that lead to a better outcome
- Determine the best policy – that with greatest expected reward over all possible (or reasonable) state sequences
- In dialogue systems, the reward is delayed – the only supervised information comes at the end of the dialogue
Types of Reinforcement Learning

• **Passive**
  – The policy is fixed, the goal is to learn how good the policy is
  – The agent executes a set of trials in the environment using its policy $\pi$

• **Active**
  – The agent learns a complete model with outcome probability for all actions
  – the learner actively chooses actions to gather experience $\rightarrow$ exploration vs exploitation
Q-learning – Learning an Action-Utility Function

- **Q(s,a)** – the value of doing action $a$ in state $s$
  \[ U(s) = \max_a Q(s, a) \]

- **Expected cumulative reward** $Q(s,a)$ for taking action $a$ in state $s$ can be computed by the **Bellman equation**:
  \[ Q(s,a) = R(s) + \gamma \sum_{s'} \{ \Pr(s' | s, a) \max_{a'} Q(s', a') \} \]

  - immediate reward in state $s$
  - expected discounted utility of all possible next states $s'$, assuming once there we take optimal action $a'$
Q-learning Algorithm

- **Input**: \( \text{percept} = \) current state \( s' \) and reward \( r' \)
- **Persistent**:
  \( Q \) – a table of state-action values indexed by \( s \) and \( a \)
  \( N_{sa} \) – a table of frequencies for state-action pairs
  \( s, a, r \) – the previous state, action and reward
- **Returns an action**
  1. If \( \text{Terminal}(s) \) then \( Q(s,\text{None}) \leftarrow r' \)
  2. If \( s \) is not null then
     Increment \( N_{sa}(s,a) \)
     \[ Q(s,a) \leftarrow Q(s,a) + \alpha N_{sa}(s,a)(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \]
  3. \( s, a, r \leftarrow s', \arg\max_{a'} f(Q(s',a'), N_{sa}(s',a')), r' \)
  4. return \( a \) trade-off between greed and curiosity
Topics in Human Computer Interaction

MDP-based Dialogue Models
Information for Q-learning Algorithm

- $N_{sa}$
  - Conversations with real people (Singh et al., 2002)
    > Carefully hand-tune small number of states and policies
    > Build a dialogue system that explores the state space by generating many (random) conversations with real people
    > Set probabilities from this corpus
  - Random conversations with simulated people (Levin et al., 1997)
    > Have millions of conversations with simulated people
    > Can have a slightly larger state space

- $r'$
  - The final reward for whole dialogue $R(s_1,a_1,s_2,a_2,\ldots,s_n)$
  - Can be calculated using DS evaluation techniques
Reward Examples

- Number of user interactions
- Number of corrections
- Number of database accesses
- Speech recognition errors
- Dialogue duration
+ Successful dialogue
± User satisfaction measures
Example – NJFun (Singh et al., 2002)

| S1: Welcome to NJFun. How may I help you?          | U1: I’d like to find um winetasting in Lambertville in the morning.  |
| Did you say you are interested in Lambertville?   | (ASR: I’d like to find out wineries the in the Lambertville in the morning) |
| Did you say you want to go in the morning?        | Yes.                                                               |
| I found a winery near Lambertville that is open in the morning. It is [...]. Please give me feedback by saying “good”, “so-so”, or “bad”. | Yes.                                                               |
| Good                                              |                                                                    |
NJFun

- **Obtains three attributes**
  - Activity type
  - Location
  - Time

- **Basic strategy**
  - first get activity type, then location, finally time
  - then make a database query
System Choices

• Initiative
  – User: open questions with an unrestricted grammar
  – System: directed prompts with a restricted grammar
  – Mixed: directed prompts with unrestricted grammar

• Confirmation
  – ExpConf: asks the user to verify an attribute
  – NoConf: no confirmation
### State Space

<table>
<thead>
<tr>
<th>G</th>
<th>A</th>
<th>C</th>
<th>V</th>
<th>T</th>
<th>M</th>
<th>H</th>
</tr>
</thead>
</table>

- **Greet (G)** – the system has greeted the user (0,1)
- **Attribute (A)** – the attribute the system is trying to obtain or verify (1=activity, 2=location, 3=time, 4=done)
- **Confidence (C)** – ASR confidence (0,1,2,3,4)
- **Value (V)** – the system has obtained a value for the current attribute (0,1)
- **Tries (T)** – number of times the system has asked for the attribute value (0,1,2)
- **Grammar (M)** – type of grammar most recently used (0=restrictive, 1=non-restrictive)
- **History (H)** – previous interaction was problematic (0,1)
# Action Choices

<table>
<thead>
<tr>
<th>Action</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>GreetS</td>
<td>Welcome to NJFun. Please say an activity name or say ‘list activities’ for a list of activities I know about.</td>
</tr>
<tr>
<td>GreetU</td>
<td>Welcome to NJFun. How may I help you?</td>
</tr>
<tr>
<td>ReAsk1S</td>
<td>I know about amusement parks, aquariums, cruises, historic sites, museums, parks, theaters, wineries and zoos. Please say an activity name from this list.</td>
</tr>
<tr>
<td>ReAsk1M</td>
<td>Please tell me the activity type. You can also tell me the location and time.</td>
</tr>
<tr>
<td>Ask2S</td>
<td>Please say the name of the town or city that you are interested in.</td>
</tr>
<tr>
<td>Ask2U</td>
<td>Please give me more information.</td>
</tr>
<tr>
<td>ReAsk2S</td>
<td>Please tell me the name of the town or city that you are interested in.</td>
</tr>
<tr>
<td>ReAsk2M</td>
<td>Please tell me the location that you are interested in. You can also tell me the time.</td>
</tr>
<tr>
<td>Ask3S</td>
<td>What time of the day do you want to go?</td>
</tr>
<tr>
<td>ReAsk3S</td>
<td>Do you want to go in the morning, in the afternoon, or in the evening?</td>
</tr>
</tbody>
</table>
Policy Class (Excerpt) – Exploratory for Initiative and Confirmation (EIC)

<table>
<thead>
<tr>
<th>Choice-States</th>
<th>Action Choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>G A C V T M H</td>
<td>GreetS,GreetU</td>
</tr>
<tr>
<td>0 1 0 0 0 0 0</td>
<td>ReAsk1S,ReAsk1M</td>
</tr>
<tr>
<td>1 1 0 0 1 0 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 0 1 0 0 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 0 1 0 1 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 1 0 0 0 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 1 1 0 0 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 1 1 0 1 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 2 1 0 0 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 1 1 0 1 0</td>
<td>NoConf,ExpConf1</td>
</tr>
<tr>
<td>1 1 2 1 0 1 0</td>
<td>ReAsk1S,ReAsk1M</td>
</tr>
<tr>
<td>1 1 4 0 0 0 0</td>
<td>ReAsk1S,ReAsk1M</td>
</tr>
<tr>
<td>1 1 4 0 1 0 0</td>
<td>ReAsk1S,ReAsk1M</td>
</tr>
</tbody>
</table>
Example (I)

- **Initial state:**

<table>
<thead>
<tr>
<th>Choice States</th>
<th>Action Choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>G  A  C  V  T  M  H</td>
<td>GreetS, GreetU</td>
</tr>
<tr>
<td>0  1  0  0  0  0  0</td>
<td></td>
</tr>
</tbody>
</table>

- **Possible actions:**

  **GreetU:** Welcome to NJFun. How may I help you?

  **GreetS:** Welcome to NJFun. Please say an activity name or say “list activities” for a list of activities I know about

  Start with **GreetU**
Example (II)

S1: Welcome to NJFun. How may I help you?
U1: I’d like to find um wine tasting in Lambertville in the morning.

(ASR: I’d like to find out wineries the in the Lambertville in the morning)

<table>
<thead>
<tr>
<th>Choice States</th>
<th>Action Choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>G  A  C  V  T  M  H</td>
<td>ExpConf, NoConf</td>
</tr>
<tr>
<td>1  1  2  1  0  0  0</td>
<td></td>
</tr>
</tbody>
</table>

Possible actions:
**ExpConf:** Did you say you are interested in wineries?
**NoConf:** say nothing move directly to the next state  
Choose **NoConf**
Example (III)

S1: Welcome to NJFun. How may I help you?
U1: I’d like to find um winetasting in Lambertville in the morning.
   *(ASR: I’d like to find out wineries the in the Lambertville in the morning)*
S2: Did you say you are interested in Lambertville?
U2: Yes.
S3: Did you say you want to go in the morning?
U3: Yes.
S4: I found a winery near Lambertville that is open in the morning. It is [...].
   Please give me feedback by saying “good”, “so-so”, or “bad”.
U4: Good

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>Turn</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>A</td>
<td>C</td>
<td>V</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Learning an Optimal Dialogue Policy

• **Training**
  – Corpus of experimental dialogues – 54 participants
    > a set of six tasks → 311 dialogues
    > participants rated the system on a number of measures
  – DS chooses randomly between available actions in any state where there is a choice of actions
  – The reward function was based on a *binary completion* measure:
    \[ \begin{cases} 
    1 & \text{if the database is queried with the correct attributes} \\
    0 & \text{otherwise} 
    \end{cases} \]
  – The optimal dialogue policy was computed

• **Testing**
  – 21 participants tested the system
Evaluation – User Survey

1. Please give your feedback for this conversation (good, so-so, bad)

2. Did you complete the task and get the information you needed? (yes, no)

3. In this conversation,
   a. it was easy to find the place I wanted
   b. I knew what I could say at each point in the dialogue
   c. NJFun understood what I said

4. Based on my current experience with NJFun, I’d use NJFun regularly to find a place to go when I’m away from my computer
The Learned Policy

• **The best use of initiative is**
  – begin with user initiative
  – back off to either mixed or system initiative if it was necessary to re-prompt for an attribute

• **The specific back-off method differs according to the attribute involved**

• **The best confirmation strategy is to confirm at lower acoustic confidence levels**
## Results: Training versus Optimal Policy

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
<th>Statistical significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary completion</td>
<td>52%</td>
<td>64%</td>
<td>p=0.059 (trend)</td>
</tr>
<tr>
<td>Weak completion</td>
<td>1.72</td>
<td>2.18</td>
<td>p=0.029</td>
</tr>
<tr>
<td>Survey measures</td>
<td></td>
<td></td>
<td>No difference (trend towards the middle)</td>
</tr>
</tbody>
</table>
Problems with the MDP Model

• **Large state space**
  – much of dialogue content and dialogue history is ignored

• **The system state is often incorrect**

• **No principled way to handle N-best ASR output**

• **Solution:**
  – Represent the dialogue process using a POMDP (Partially Observable Markov Decision Process)
Dialogue as POMDP

- System belief state maintains multiple dialogue hypotheses
- System actions are based on full belief-state distribution
- N-best ASR/SLU inputs included in the belief state
  - Can proceed even if confidence for ASR/SLU is low
- If a misunderstanding is detected, the system doesn’t need to backtrack
- Belief monitoring
  - Belief distribution is re-calculated each time a new observation is received
Reading Material

• **Bayesian networks**
  – A computational architecture for conversation, Horvitz and Paek (1999)
  – Conversation as action under uncertainty, Paek and Horvitz (2000)

• **Markov decision processes**
  – Optimizing dialogue management with RL: Experiments with the NJFun system, Singh *et al.* (2002)