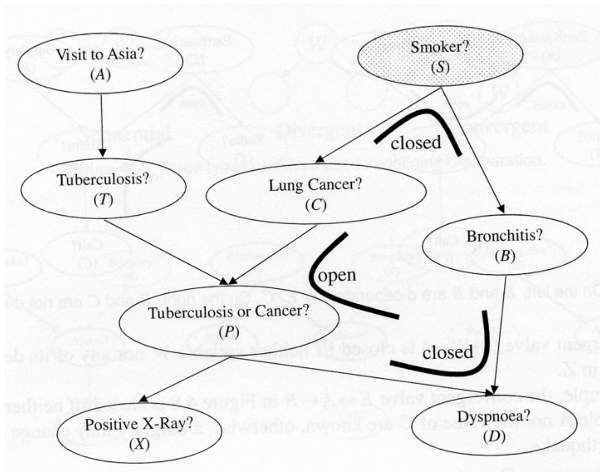
Spezielle Themen der Künstlichen Intelligenz

11.Termin:

Dynamic Bayesian Networks

Rational Decision-Making under Uncertainty



Using Bayes Nets to model change

The world changes; we need to track and predict it

Diabetes management vs vehicle diagnosis

Basic idea: copy state and evidence variables for each time step

$\mathbf{X}_t$ = set of unobservable state variables at time t
e.g., $BloodSugar_t$, $StomachContents_t$, etc.

$\mathbf{E}_t$ = set of observable evidence variables at time t
e.g., $MeasuredBloodSugar_t$, $PulseRate_t$, $FoodEaten_t$

This assumes **discrete time**; step size depends on problem

Notation: $\mathbf{X}_{a:b} = \mathbf{X}_a, \mathbf{X}_{a+1}, \ldots, \mathbf{X}_{b-1}, \mathbf{X}_b$

Markov chains / Markov processes

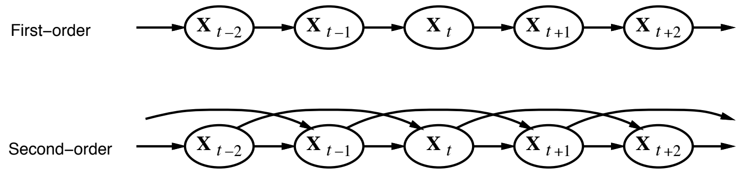

1856-1922

Construct a Bayes net from these variables: parents?

Markov assumption: $\mathbf{X}_t$ depends on **bounded** subset of $\mathbf{X}_{0:t-1}$

First-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
Second-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$

transition model



Sensor Markov assumption: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ *sensor model*

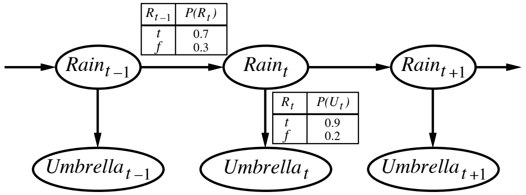
Stationary process: transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$ and sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ fixed for all t *(changes follow fixed laws)*

Markov processes

Need prior probability $P(X_0)$ over states at time 0

Then we have: $P(X_0,X_1,\ldots,X_t,E_1,\ldots,E_t)=P(X_0)\prod_{i=1..t} P(X_i \mid X_{i-1})P(E_i \mid X_i)$

Example:



First-order Markov assumption not exactly true in real world!

Possible fixes:

1. **Increase order** of Markov process
2. **Augment state**, e.g., add $Temp_t$, $Pressure_t$

Inference tasks

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
belief state—input to the decision process of a rational agent

Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$
evaluation of possible action sequences;
like filtering without the evidence

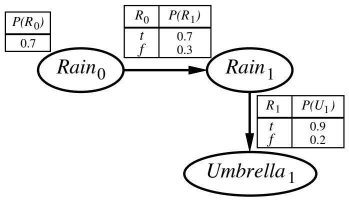
Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$
better estimate of past states, essential for learning

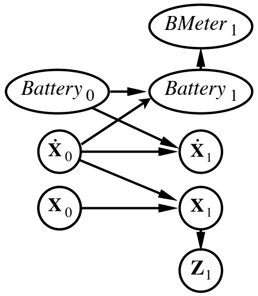
Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
speech recognition, decoding with a noisy channel
given observations, find sequence of states that is
most likely to have generated them
(e.g. Viterbi algorithm)

(see Russell & Norvig, Sect. 15.2 for algorithms)

Dynamic Bayesian networks

$\mathbf{X}_t$, $\mathbf{E}_t$ contain arbitrarily many variables in a replicated Bayes net





Need to give transition and senor model (stationary) only for first slice

↓

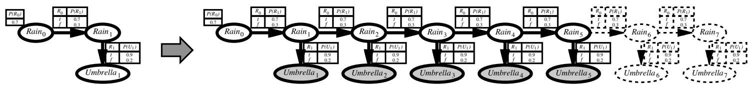
Sensor variables Z_t, $BMeter_t$;
state variables X_t, $Xdot_t$, $Battery_t$

Hidden Markov Model = DBN with a single discrete state variable

Exact inference in DBNs

DBNs are Bayesian networks, i.e., we can use our known algorithms

Exact inference: Unroll network to accommodate all observations and run exact inference algorithm (e.g. variable elimination)



Problem: inference cost for each update grows with t

Rollup filtering: add slice $t+1$, "sum out" slice t using variable elimination

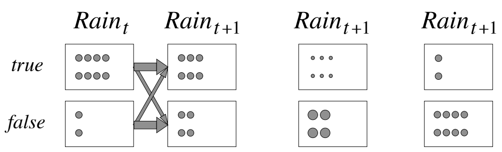
Costs (factor size) almost always exponential in number of state variables
Largest factor is $O(d^{n+1})$

Approximative inference in DBNs

Particle filtering: ensure a population of samples ("particles") that tracks the high-likelihood regions of the state-space

<u>Algorihm</u>: create N samples from prior distribution $P(X_0)$, then cycle...

1. *Propagate* by sampling next state x_{t+1}, given x_t and using $P(X_{t+1}|x_t)$
2. *Weight* samples by likelihood it assigns to new evidence $P(e_{t+1}|x_{t+1})$
3. *Resample* new N samples from the current population, probability that sample is replicated proportional to its „weight" (no. samples)



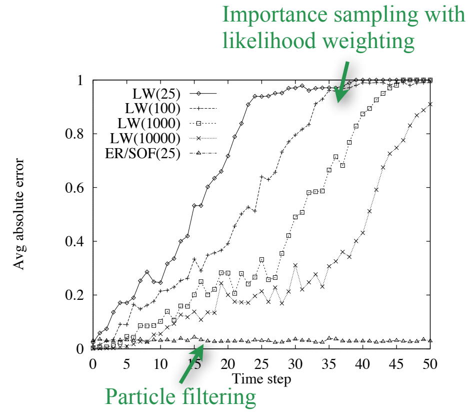
(a) Propagate (b) Weight (c) Resample

¬ umbrella observed

Particle filtering

widely used for tracking nonlinear systems, especially in vision, self-localization or mapping in mobile robots

- consistent (proof see Sect. 15.5)
- approximation error remains bounded over time, at least empirically
- in practice efficient, yet no theoretical guarantees (so far)



Decision-making

Let action A_t = *leave for airport t minutes before flight*

<u>Question</u>: Will A_t get me there on time?

Logical agent would be unable to act rationally:

- A_{90} will get me there on time *if* there's no accident on the bridge *and* it doesn't rain *and* my tires remain intact *and* .....
  - plan success not inferrable (qualification problem)

Probability of facts relates them to own state of knowledge

- degree of belief, e.g., Pr(A_{25} | no reported accidents) = 0.06
- changes as new (soft or hard) evidence comes in

Decision-making

Degree of belief cannot account for decision-making alone

- suppose the agent believes the following:
  - Pr(A_{25} gets me there on time | …) = 0.04
  - Pr(A_{90} gets me there on time | …) = 0.70
  - Pr(A_{120} gets me there on time | …) = 0.95
  - Pr(A_{1440} gets me there on time | …) = 0.999

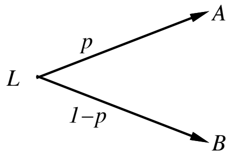
<u>Instead</u>: rational decision-making must depend on *both*

- likelihood that goals can be achieved to a necessary degree
- relative importance of goals
  - modeled as preferences for possible outcomes (risks, costs, rewards, etc.),
  - represented using utility theory

decision theory = probability theory + utility theory

Basis of utility theory - preferences

An agent chooses among prizes (A, B, etc.) and lotteries, i.e., situations with uncertain prizes



Lottery $L = [p, A;\ (1-p), B]$

$L=[p_i,C_i]$ ↔ outcome C_i can occur with probability p_i

Notation:

$A \succ B$	A preferred to B
$A \sim B$	indifference between A and B
$A \succsim B$	B not preferred to A

Key question: how are preferences related when making decisions?

Rational preferences

Idea: preferences of a rational agent must obey constraints.
Rational preferences $\Rightarrow$
behavior describable as maximization of expected utility

Constraints:

Orderability
$(A \succ B) \vee (B \succ A) \vee (A \sim B)$ — *Agent cannot avoid deciding*

Transitivity
$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$

Continuity
$A \succ B \succ C \Rightarrow \exists p\ [p, A;\ 1-p, C] \sim B$ — *Indifferent between lottery A vs. C, and getting B for sure*

Substitutability
$A \sim B \Rightarrow [p, A;\ 1-p, C] \sim [p, B; 1-p, C]$ — *Lotteries with comparable prizes comparable*

Monotonicity
$A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A;\ 1-p, B] \succsim [q, A;\ 1-q, B])$

Rational preferences contd.

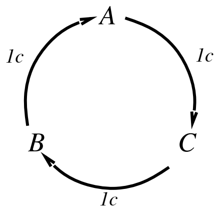
Violating the constraints leads to self-evident irrationality

For example: an agent with intransitive preferences can be induced to give away all its money

If $B \succ C$, then an agent who has C would pay (say) 1 cent to get B

If $A \succ B$, then an agent who has B would pay (say) 1 cent to get A

If $C \succ A$, then an agent who has A would pay (say) 1 cent to get C



Utilities and preferences

Preferences are a basic property of rational agents. The existence of a utility function follows then from two principles:

Theorem (Ramsey, 1931; von Neumann and Morgenstern, 1944):
Given preferences satisfying the constraints
there exists a real-valued function U such that

$U(A) \geq U(B) \quad \Leftrightarrow \quad A \succsim B$ — Utility principle

$U([p_1, S_1;\ \ldots\ ;\ p_n, S_n]) = \Sigma_i\ p_i U(S_i)$ — Max. Expected Utility principle

MEU principle:
Choose the action that maximizes expected utility

That is, a utility function can be formulated in accord with agent's preferences
Can act rationally without *explicitely* trying to maximize expected utility.

Utility functions

Utility function maps from states to real number. But which numbers?

Preferences of real agents are usually systematic, and there are systematic ways of designing utility functions.

Monotonic preferences: Agent prefers more money to less, all other things being equal. Does that say anything about lotteries involving money?

Get \$1.000.000 for sure of flip coin for 50% chance of getting \$3.000.000?

Expected monetary value (EMV) = 0.5 \$0 + 0.5 \$3.000.000 = \$1.500.00

EU(Accept) = 0.5 $U(S_{k+0})$ *+ 0.5* $U(S_{k+3.000.000})$ (S_k=state of possessing \$k)

EU(Decline) = $U(S_{k+1.000.000})$

Rational decision depends on utilities assigned to outcome states!

Multi-attribute utility

Often outcomes are characterized by two or more attributes.

How can we handle utility functions of many variables $X_1 \ldots X_n$?
E.g., what is $U(Deaths, Noise, Cost)$?

How can complex utility functions be assessed from preference behaviour?

Idea 1: identify conditions under which decisions can be made without complete identification of $U(x_1, \ldots, x_n)$ (exploiting the **dominance** of x_i)

Idea 2: identify various types of **independence** in preferences and derive consequent canonical forms for $U(x_1, \ldots, x_n)$

Recap‘

A decision will lead to new states with values (prizes) or lotteries (situations with uncertain prizes).

Rational agents have constrained preferences over values

Given preferences satisfying the constraints there exists a real-valued function U such that

$$U(A) \geq U(B) \quad \Leftrightarrow \quad A \succsim B$$
$$U([p_1, S_1; \ \ldots \ ; \ p_n, S_n]) = \Sigma_i \ p_i U(S_i)$$

A utility function can be formulated in accord with agent‘s preferences.

Decision-making

Principle of maximum expected utility (MEU)
An agent is rational **iff** it chooses the action that yields the highest expected utility, averaged over all possible outcomes of the action

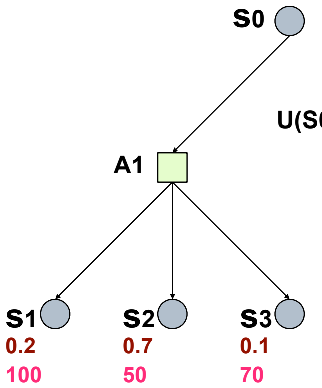
<u>*Note:*</u> a non-deterministic action can have several outcomes $Result_i(A)$

Prior to executing A, the agent needs to...

1. determine the probabilities $P(Result_i(A)|Do(A),E)$
2. calculate the expected utility of A, given evidence E:
   $EU(A|E)=\sum_i P(Result_i(A)|Do(A),E) \ U(Result_i(A))$
   with $U(S)$ utility function of state S
3. decide which action to take

Let‘s focus on „one-shot decisions“ over single actions first

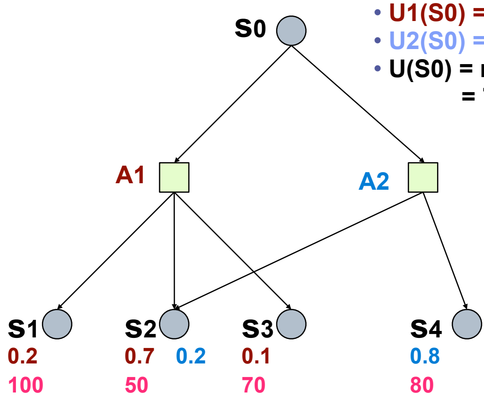
One State/One Action Example



U(S0) = 100 x 0.2 + 50 x 0.7 + 70 x 0.1
= 20 + 35 + 7
= 62

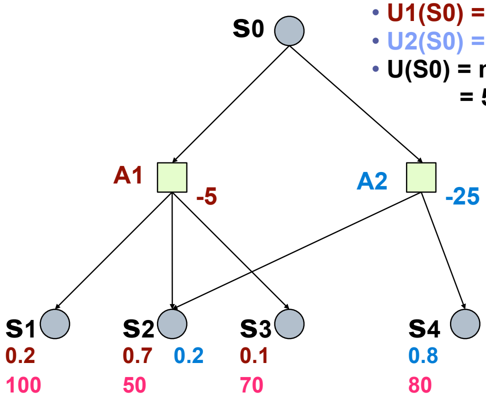
One State/Two Actions Example

- U1(S0) = 62
- U2(S0) = 74
- U(S0) = max{U1(S0),U2(S0)}
  = 74



Introducing Action Costs

- U1(S0) = 62 – 5 = 57
- U2(S0) = 74 – 25 = 49
- U(S0) = max{U1(S0),U2(S0)}
  = 57



MEU Principle

The MEU is the basis of the field of decision theory. It provides a normative criterion for rational choice of action

A rational agent must have complete model of *Actions*, *Utilities*, *States*

- even if you had a complete model, decision making becomes computationally intractable

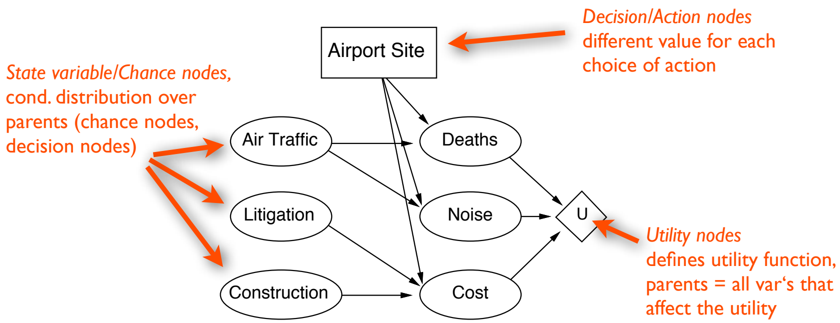
In fact, a truly rational agent takes into account the *utility/costs of reasoning* as well -- bounded rationality (Simon, 1957)

Nevertheless, great progress has been made and we are able to solve much more complex decision-theoretic problems than ever before

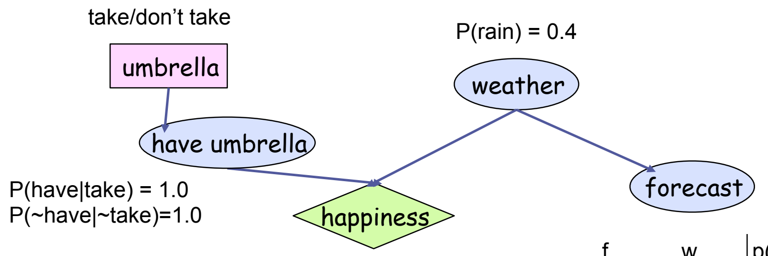
Bayesian Decision Networks

Extend Bayesian Networks to handle actions and utilities

- also called influence diagrams



Umbrella Network

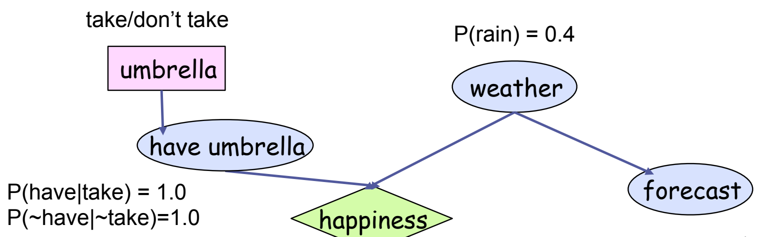


U(have,rain) = -25
U(have,~rain) = 0
U(~have, rain) = -100
U(~have, ~rain) = 100

f	w	p(f|w)
sunny	rain	0.3
rainy	rain	0.7
sunny	no rain	0.8
rainy	no rain	0.2

Decision Making: Umbrella Network

Queries: Should I take my umbrella?
What is the value of knowing the weather forecast?



U(have,rain) = -25
U(have,~rain) = 0
U(~have, rain) = -100
U(~have, ~rain) = 100

f	w	p(f|w)
sunny	rain	0.3
rainy	rain	0.7
sunny	no rain	0.8
rainy	no rain	0.2

Evaluating Decision Networks

Use Bayesian Network inference methods to compute expected utilities.

Algorithm:

Set the evidence variables for current state

For *each possible value* of a decision node:

1. set decision node to that value
2. calculate the posterior probability of the parent nodes of the utility node, using BN inference
3. calculate the resulting (expected) utility for action

Return the action with the highest utility

Question often: what do I need to know to make a solid decision?

Value of information

Idea: compute value of acquiring each possible piece of evidence
Can be done **directly from decision network**

Example: buying oil drilling rights
- Two blocks A and B, exactly one has oil, worth k
- Prior probabilities 0.5 each, mutually exclusive
- Current price of each block is $k/2$
- "Consultant" offers accurate survey of A. Fair price?

Solution: compute expected value of information
= expected value of best action given the information
minus expected value of best action without information

Survey may say "oil in A" or "no oil in A", **prob. 0.5 each** (given!)
= [$0.5 \times$ value of "buy A" given "oil in A"
+ $0.5 \times$ value of "buy B" given "no oil in A"]
– 0
= $(0.5 \times k/2) + (0.5 \times k/2) - 0 = k/2$

CITEC 29 Sociable Agents

General formula

Current evidence E, current best action α
Possible action outcomes S_i, potential new evidence E_j

$$EU(\alpha|E) = \max_a \Sigma_i\ U(S_i)\ P(S_i|E,a)$$

Suppose we knew $E_j = e_{jk}$, then we would choose $\alpha_{e_{jk}}$ s.t.

$$EU(\alpha_{e_{jk}}|E, E_j = e_{jk}) = \max_a \Sigma_i\ U(S_i)\ P(S_i|E,a,E_j = e_{jk})$$

E_j is a random variable whose value is *currently* unknown
$\Rightarrow$ must compute expected gain over all possible values:

$$VPI_E(E_j) = \left(\Sigma_k\ P(E_j = e_{jk}|E) EU(\alpha_{e_{jk}}|E, E_j = e_{jk})\right) - EU(\alpha|E)$$

(VPI = value of perfect information)

CITEC 30 Sociable Agents

Information-gathering agent

```
function INFORMATION-GATHERING-AGENT(percept) returns eine Aktion
   static: D, ein Entscheidungsnetzwerk

   integriere percept in D
   j ← der Wert, der WPI(E_j) − Kosten(E_j) maximiert
   if WPI(E_j) > Kosten(E_j)
      then return REQUEST(E_j)
   else return die beste Aktion aus D
```

Agent chooses between sensing action (*REQUEST*, which will yield evidence in next percept) or „real action"

Extension: consider all possible sensing action sequences and all possible outcomes of those requests. Because values of requests depend on previous requests, need to build conditional plans

CITEC 31 Sociable Agents