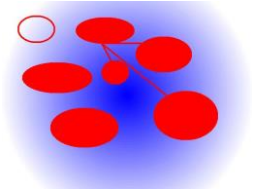


CS6462

Probabilistic and Explainable AI

Lesson 7

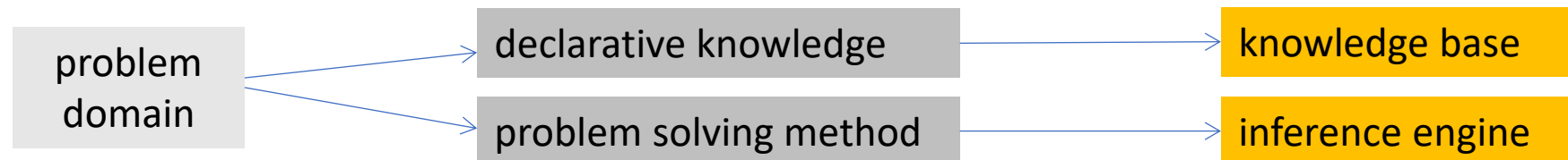
Model-Based Reasoning

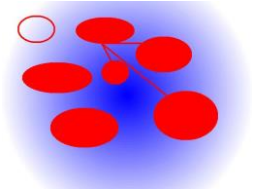


Models in AI

Definition (recall):

- a construct designed to respond in the same way as the system we would like to understand
- numerical models emulate stochasticity, i.e. using pseudorandom number generators, to simulate actually random phenomena and other uncertainties
- models produce *observations* we can measure in the real world
- formal models are a precise statement of components to be used and the relationships among them





Models in AI (cont.)

Model denotation:

- formal models - typically denoted mathematically

Example: a beta-Bernoulli model

formal functions (produce distribution models)

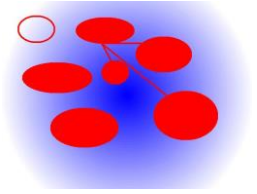
- statistical model, for generating a coin flip from a potentially biased coin

$\mathbf{x} \approx \mathbf{Beta}(\boldsymbol{\alpha}, \boldsymbol{\beta})$, \mathbf{x} is a latent variable (the bias of the coin)

$\mathbf{y} \approx \mathbf{Bernoulli}(\mathbf{x})$, \mathbf{y} is the value of the flipped coin

Targets and Methods:

- capture instantaneous behavior, temporal behavior, structure
- methods: diagnosis, decision making, prediction, planning



Model-Based Reasoning

Definition:

- refers to inference methods used in expert systems based on a model of the physical world
- models used as a basis for fault detection and diagnosis - often referred to as *model based reasoning*

Advantages:

- models can be used for prediction of the impacts of faults as well as diagnosis
- application development is **formalized** - easier to check and reuse
- assumptions and limitations are likely to be cleared
- models are likely to reflect science rather than observed coincidences

Variations:

- normal operations or abnormal operations
- static vs. dynamic
- causal or non-causal
- compiled vs. first principles
- probabilistic vs. deterministic

quantitative - based on numbers and equations
and/or
qualitative - based on cause/effect

Models of Abnormal vs. Normal Operation



Application domain:

- systems characterized by numerical variables

Models of normal operation:

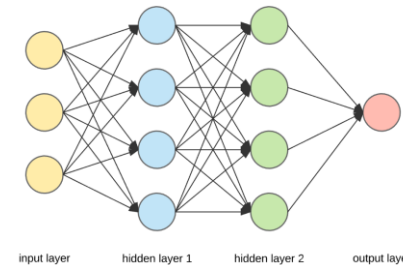
- algebraic equations, differential equations, neural nets, state transition diagrams

Fault detection:

- involves checking that the models are being followed in the observed sensor data

Models of abnormal operation:

- engineering models of normal behavior - complex and costly to develop and maintain
- alternative - develop models of abnormal behavior
- ultimately link root cause problems to observable symptoms



Example 1: modeling with equations

Example 2: modeling with state transition diagram

Static vs. Dynamic Models



Definition:

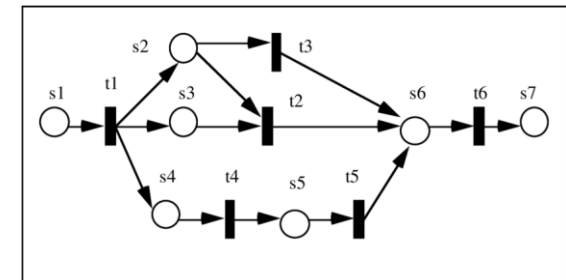
- dynamic models explicitly model behavior over time, while static models do not
- dynamic – traditionally quantitative models
- static – both quantitative and qualitative models
- static models provide a 'snapshot' of a system's response to a set of input conditions

Dynamic model example:

- state transition diagrams or Petri nets - model changes in state that occur over time

Static model example:

- a set algebraic equations



Causal Models

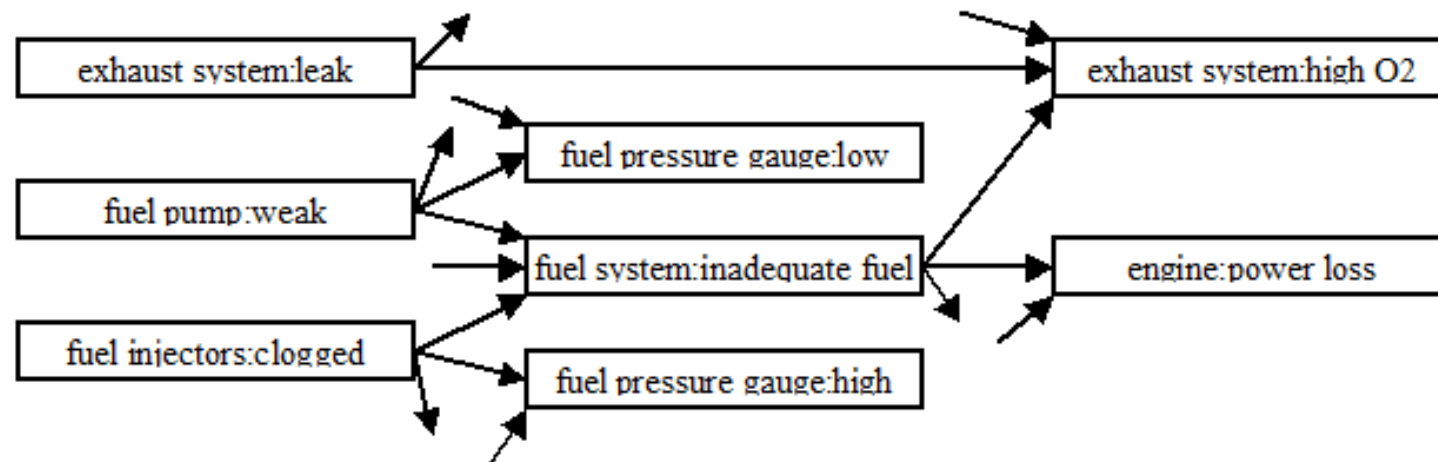


Definition:

- a qualitative model form using binary variables, usually represented as a set of variables depicted as nodes in a directed graph

Example:

- a part of a cause/effect model of problems for vehicles
- boxes represent problems; links represent cause and effect



Compiled vs. First Principles Models

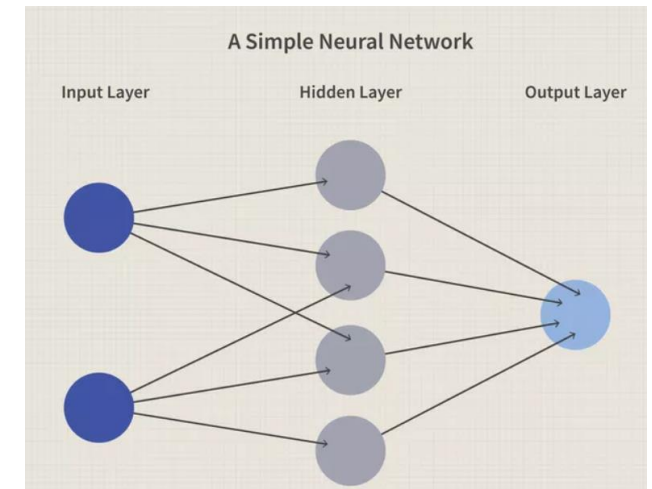


First Principles models:

- based on science laws or device implementation knowledge, rather than on data
- engineering design models, qualitative models such as causal fault propagation models, or state transition diagrams
- often referred to as using “deep knowledge”

Compiled models:

- simplified versions of more complex models
- empirical models - derived directly from data
 - use data derived from tests or from the output of simulation models
 - examples: regression models, neural nets
 - disadvantage: need to be rebuilt when there are changes in the data model



Probabilistic vs. Deterministic Models



Uncertainty:

- lack of exact knowledge, regardless of what is the cause of this deficiency
- examples: imperfect observations; inconsistency caused by timing; imperfect models; missing observations

Deterministic models:

- assume certainty in all aspects
- examples: timetables, pricing structures, a linear programming model, maps, etc.

Probabilistic (Stochastic) models:

- represent uncertainty
- examples: queueing models, markov chains, most simulations, etc.

Uncertainty – Classification



Stems from various sources (classification by Regan et al.):

- inherent randomness – uncertainty about the outcome; can be quantified
- measurement error – causes uncertainty about the value of the measured quantity; can be estimated by statistical methods, if several samples are taken
- systematic error – error in the measurements resulting from a bias in the sampling; difficult to quantify
- natural variation – real systems change in time and place, and so do the parameters
- model uncertainty – models are abstractions of the real systems; less important variables and interactions can be left out; the formal functions are abstractions of the real processes; needed careful consideration of the range of possible values and their probabilities;
- subjective judgment - occurs due to interpretation of data or behavior

H.M. Regan, M. Colyvan and M.A. Burgman. A taxonomy and treatment of uncertainty for ecology and conservation biology. Ecol. Appl., 12 (2002), pp. 618-628

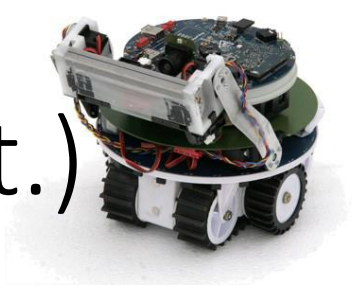
Probabilistic Model Example: POMDP



- *Example: Awareness Self-Initiation for marXbot*
 - marXBot – swarm robotics platform
 - a behavior model based on the *Partially Observable Markov Decision Processes (POMDP)* [Litt_1996]
 - appropriate when there is uncertainty and lack of information needed to determine the state of the entire swarm
 - SC (service component):
 - takes as input *observable* situations, involving other swarm robots and the environment
 - generates as output actions initiating robot activity
 - the generated actions affect the state of the ensemble

M. L. Littman, Algorithms for Sequential Decision Making, PhD Thesis, Department of Computer Science, Brown University, 1996.

Probabilistic Model Example: POMDP(cont.)



Case Study – Swarm of Robots:

- **marXbot*** - a modular research robot equipped with a set of devices to interact with other robots of the swarm or the robotic environment
- marXbots are able to work in teams where they coordinate based on simple interactions on group tasks

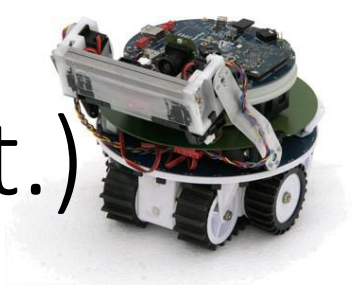
Example:

*a group of marXbots may collectively move a relatively heavy object from point **A** to point **B** by using their grippers*



*ULB (Université Libre de Bruxelles)

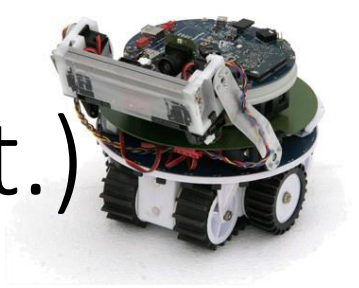
Probabilistic Model Example: POMDP(cont.)



Formal Model – a tuple $\mathbf{M} = \langle \mathbf{S}; \mathbf{A}; \mathbf{T}; \mathbf{O}; \mathbf{R}; \mathbf{Z} \rangle$:

- \mathbf{S} is a finite set of states;
- An initial belief state $\mathbf{s}_0 \in \mathbf{S}$ is based on $\mathbf{p}_0(\mathbf{s}_0; \mathbf{s}_0 \in \mathbf{S})$ - a discrete probability distribution over the set of states \mathbf{S} (for each state the robot's belief that is currently occupying that state).
- \mathbf{A} is a finite set of actions that may be undertaken by the robot.
- $\mathbf{T}: \mathbf{S} \times \mathbf{A} \rightarrow \Pi(\mathbf{S})$ is the state transition function, giving for each swarm state \mathbf{s} and robot action \mathbf{a} , a probability distribution over states. $\mathbf{T}(\mathbf{s}; \mathbf{a}; \mathbf{s}')$ computes the probability of ending in state \mathbf{s}' , given that the start state is \mathbf{s} and the robot takes action \mathbf{a} .
- $\mathbf{O}: \mathbf{A} \times \mathbf{S} \rightarrow \Pi(\mathbf{Z})$ is the observation function giving for each swarm state \mathbf{s} and robot action \mathbf{a} , a probability distribution over observations \mathbf{Z} . $\mathbf{O}(\mathbf{s}'; \mathbf{a}; \mathbf{z})$ is the probability of observing \mathbf{z} , in state \mathbf{s}' after taking action \mathbf{a} .
- $\mathbf{R}: \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{R}$ is a reward function, giving the expected immediate reward gained by the robot for taking an action in a state \mathbf{s} , e.g., $\mathbf{R}(\mathbf{s}; \mathbf{a})$. The reward is a scalar value in the range [0..1] determining, which should be undertaken by the robot in compliance with the swarm goals.

Probabilistic Model Example: POMDP(cont.)



- *Formal Model - Interpretation:*

- a marXbot swarm is currently occupying the state s = “new object to be moved is discovered, but no moving team has been formed yet and still no other marXbot has self-initiated for team formation”;
- idle marXbot ready to undertake a few actions A , including the action a = “self-initiation for team formation”;
- Reasoning steps:
 - The marXbot computes its current belief state s_0 – the robot picks up the state with the highest probability p_0 and eventually $s_0 = s$.
 - The marXbot computes the probability p_1 of the swarm occupying the state $s' =$ “new object is discovered and a marXbot has self-initiated for team formation” if the action a is undertaken from state s_0 .
 - The marXbot computes the probability $p_2(z \mid s'; a)$ of observation $z =$ “there are sufficient numbers of idle marXbots to form a new exploration team”.
 - The marXbot computes the reward $r(s_0; a)$ for taking the action a (self-initiation for team formation) in state s_0 . If no other immediate actions should be undertaken (forced by other swarm goals), the reward r should be the highest possible, which will determine the execution of action a .

Probability Computation

probability assessment - an indicator of the number of possible execution paths a robot may take, i. e., the amount of certainty (excess entropy)

Summary



Model-based reasoning:

- refers to inference methods used in expert systems based on a model of the physical world
- models used as a basis for fault detection and diagnosis

Variations:

- normal operations or abnormal operations
- static vs. dynamic
- causal or non-causal
- compiled vs. first principles
- probabilistic vs. deterministic

Probabilistic Model Example – POMDP

Next Lesson – Principles of Probabilistic Programming

Thank You!

Questions?