Reasoning Under Uncertainty Over Time

CS 486/686: Introduction to Artificial Intelligence

Outline

- Reasoning under uncertainty over time
  - Hidden Markov Models
  - Dynamic Bayes Nets

Introduction

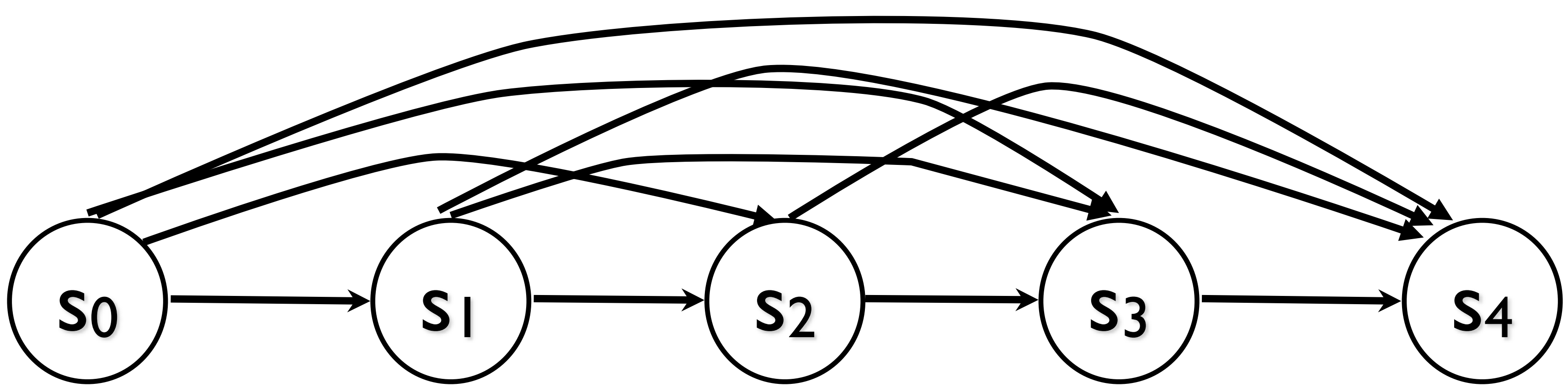
- So far we have assumed
  - The world does not change
  - Static probability distribution
- But the world does evolve over time
  - How can we use probabilistic inference for weather predictions, stock market predictions, patient monitoring, robot localization,...

Dynamic Inference

- To reason over time we need to consider the following:
  - Allow the world to evolve
  - Set of states (all possible worlds)
  - Set of time-slices (snapshots of the world)
  - Different probability distributions over states at different time-slices
  - Dynamic encoding of how distributions change over time

Stochastic Process

- Set of states: **S**
- Stochastic dynamics: $P(s_t | s_{t-1}, \ldots, s_0)$
- Can be viewed as a Bayes Net with one random variable per time-slice

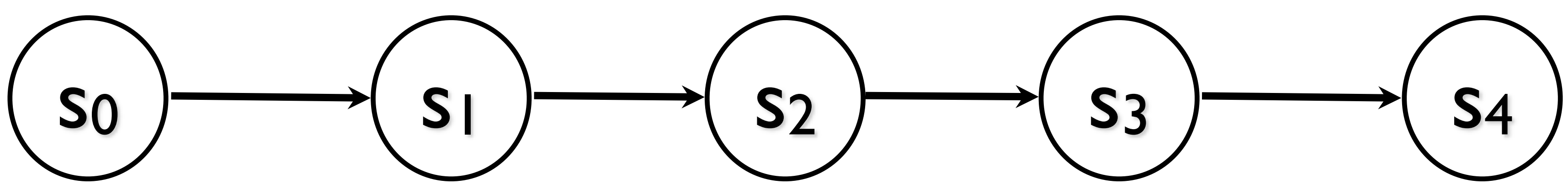


Stochastic Process

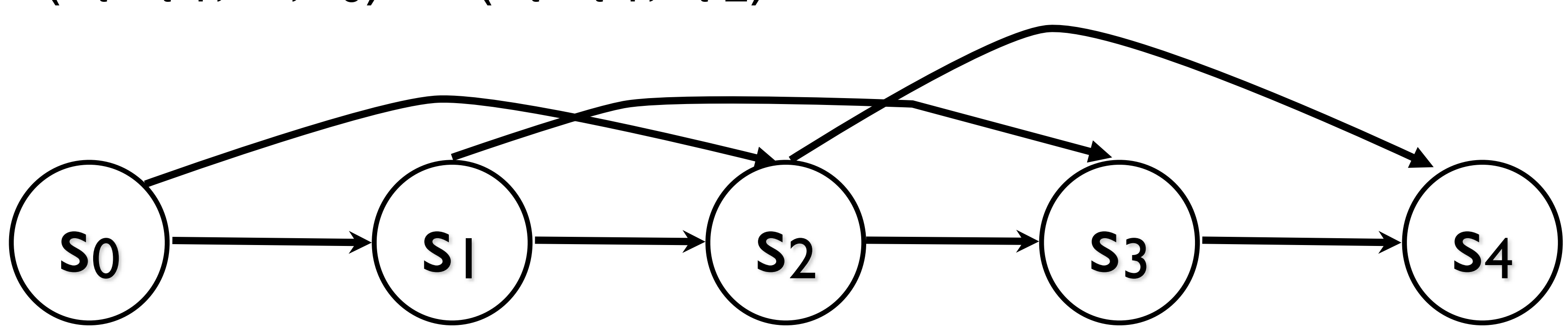
- Problems:
  - Infinitely many variables
  - Infinitely large CPTs
- Solutions:
  - **Stationary process**: Dynamics do not change over time
  - **Markov assumption**: Current state depends only on a finite history of past states

k-Order Markov Process

- Assumption: last k states are sufficient
- First-order Markov process
  - $P(s_t|s_{t-1},\ldots,s_0)=P(s_t|s_{t-1})$

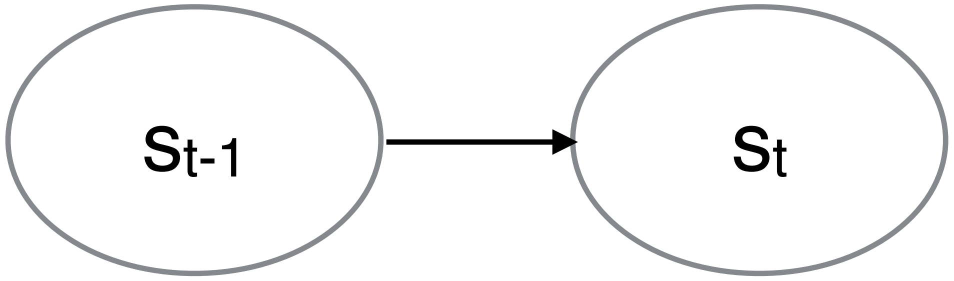


- Second-order Markov process
  - $P(s_t|s_{t-1},\ldots,s_0)=P(s_t|s_{t-1},s_{t-2})$



k-Order Markov Process

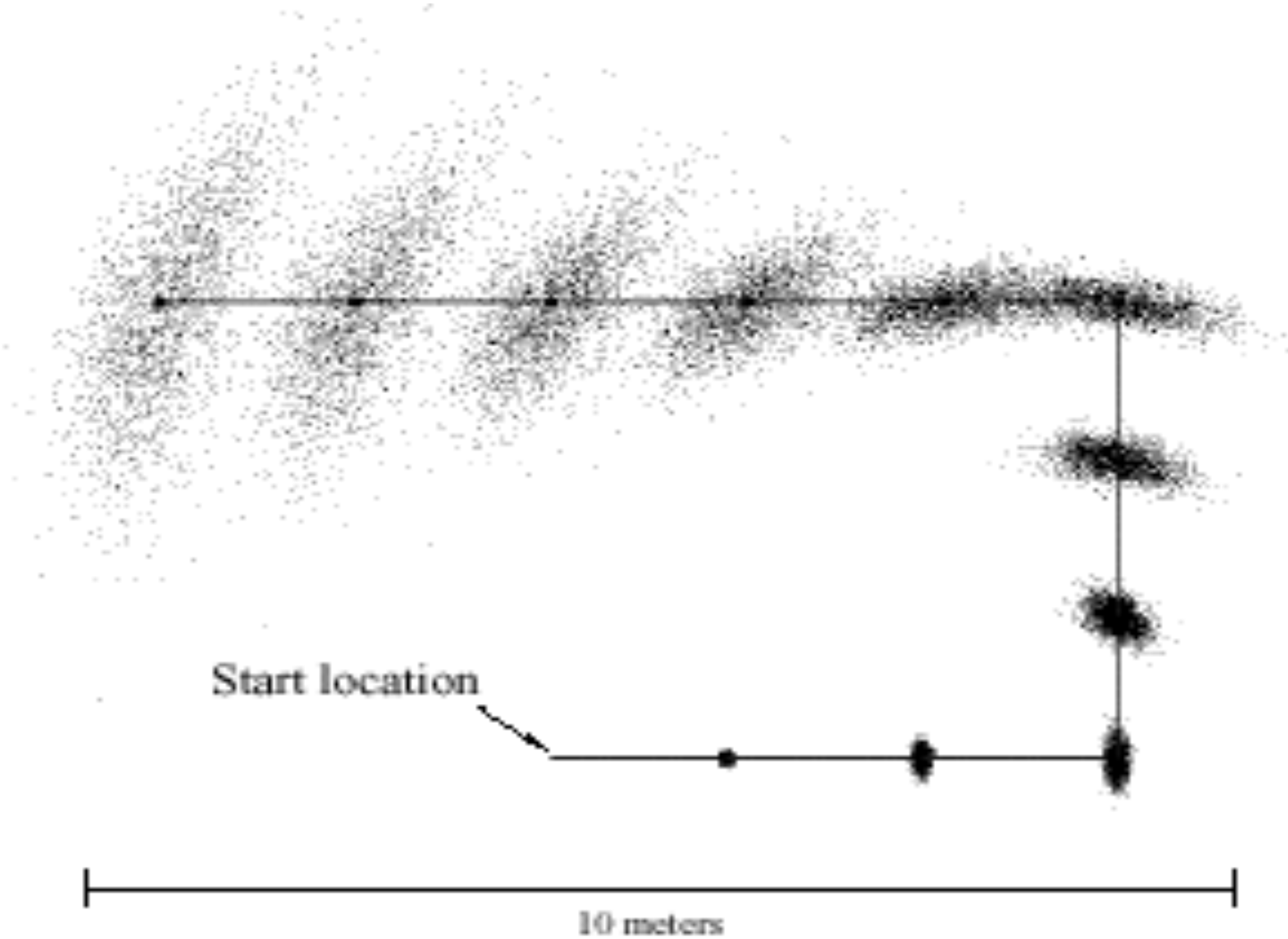
- Advantages
  - Can specify the entire process using finitely many time slices
- Example: Two slices sufficient for a first-order Markov process
  - Graph:



  - Dynamics: $P(s_t | s_{t-1})$
  - Prior: $P(s_0)$

Example: Robot Localization

- Example of a first-order Markov process



Problem: uncertainty increases over time

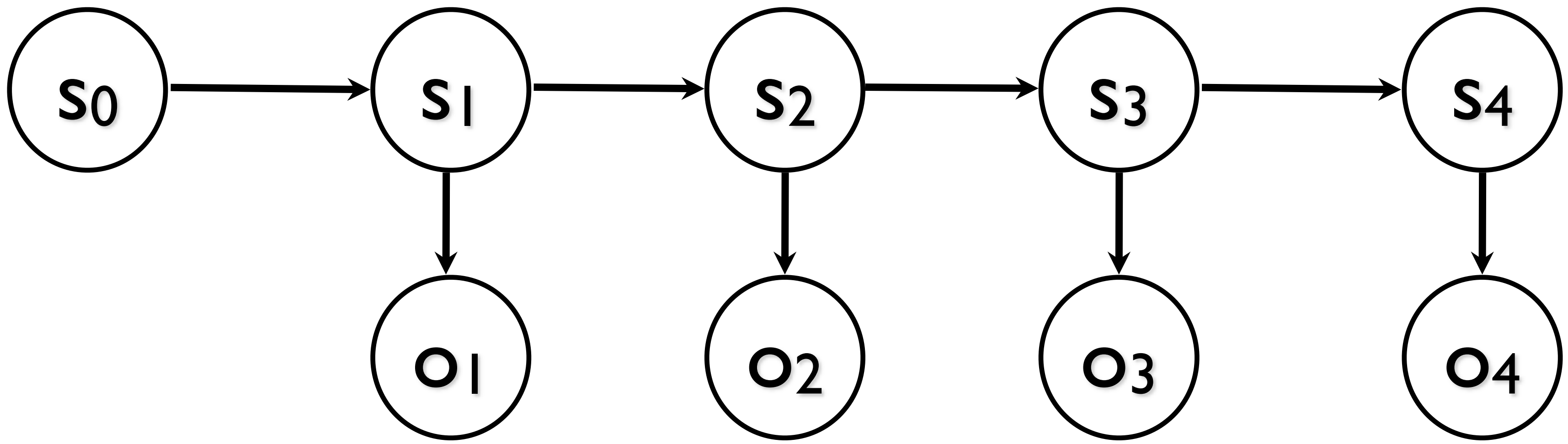
Thrun et al

Hidden Markov Models

- In the previous example, the robot could use sensors to reduce location uncertainty
- In general:
  - States not directly observable (uncertainty captured by a distribution)
  - Uncertain dynamics increase state uncertainty
  - Observations: made via sensors can reduce state uncertainty
- **Solution**: Hidden Markov Model

First Order Hidden Markov Model (HMM)

- Set of states: S
- Set of observations: O
- Transition model: $P(s_t|s_{t-1})$
- **Observation model**: $P(o_t|s_t)$
- Prior: $P(s_0)$



Example: Robot Localization

- Hidden Markov Model
  - S: (x,y) coordinates of the robot on the map
  - O: distances to surrounding obstacles (measured by laser range fingers or sonar)
  - $P(s_t|s_{t-1})$: movement of the robot with uncertainty
  - $P(o_t|s_t)$: uncertainty in the measurements provided by the sensors
- **Localization** corresponds to the query:
  - $P(s_t|o_t,...,o_1)$

Inference

- There are four common tasks
  - **Monitoring**: $P(s_t|o_t,...o_1)$
  - **Prediction**: $P(s_{t+k}|o_t,...,o_1)$
  - **Hindsight**: $P(s_k|o_t,...,o_1)$
  - **Most likely explanation**: $\text{argmax}_{st,...,s1}\ P(s_t,...,s_1|o_t,...,o_1)$
- What algorithms should we use?
  - First 3 can be done with variable elimination and the 4th is a variant of variable elimination

Monitoring

- We are interested in the distribution over current states given observations: $P(s_t|o_t,...,o_1)$
  - Examples: patient monitoring, robot localization
- Forward algorithm: corresponds to variable elimination
  - Factors: $P(s_0)$, $P(s_i|s_{i-1})$, $P(o_i|s_i)$ $1 \leq i \leq t$
  - Restrict $o_1,...,o_t$ to observations made
  - Sum out $s_0,....,s_{t-1}$
  - $\sum_{s0...st-1} P(s_0) \prod_{1 \leq i \leq t} P(s_i|s_{i-1}) P(o_i|s_i)$

Prediction

- We are interested in distributions over future states given observations: $P(s_{t+k}|o_t,\ldots,o_1)$
  - Examples: weather prediction, stock market prediction
- Forward algorithm: corresponds to variable elimination
  - Factors: $P(s_0)$, $P(s_i|s_{i-1})$, $P(o_i|s_i)$ $1 \leq i \leq t+k$
  - Restrict $o_1,\ldots,o_t$ to observations made
  - Sum out $s_0,\ldots,s_{t+k-1}, o_{t+1},\ldots,o_{t+k}$
  - $\sum_{s0\ldots st-1,ot+1,\ldots,ot+k} P(s_0) \prod_{1 \leq i \leq t+k} P(s_i|s_{i-1}) P(o_i|s_i)$

Hindsight

- Interested in the distribution over a past state given observations
  - Example: crime scene investigation
- Forward-backward algorithm: corresponds to variable elimination
  - Factors: $P(s_0)$, $P(s_i|s_{i-1})$, $P(o_i|s_i)$ $1 \leq i \leq t$
  - Restrict $o_1,\ldots,o_t$ to observations made
  - Sum out $s_0,\ldots,s_{k-1},s_{k+1},\ldots,s_t$
  - $\sum_{s0 \ldots sk-1,sk+1,\ldots st,} P(s_0) \prod_{1 \leq i \leq t} P(s_i|s_{i-1}) P(o_i|s_i)$

Most Likely Explanation

- We are interested in the most likely sequence of states given the observations: $\text{argmax}_{s0,...st}\ P(s_0,...,s_t|o_t,...,o_1)$
  - Example: speech recognition
- Viterbi algorithm: Corresponds to a variant of variable elimination
  - Factors: $P(s_0)$, $P(s_i|s_{i-1})$, $P(o_i|s_i)$ $1 \leq i \leq t$
  - Restrict $o_1,...,o_t$ to observations made
  - Max out $s_0,....,s_{t-1}$
  - $\max_{s0...st-1} P(s_0) \prod_{1 \leq i \leq t} P(s_i|s_{i-1}) P(o_i|s_i)$

Complexity of Temporal Inference

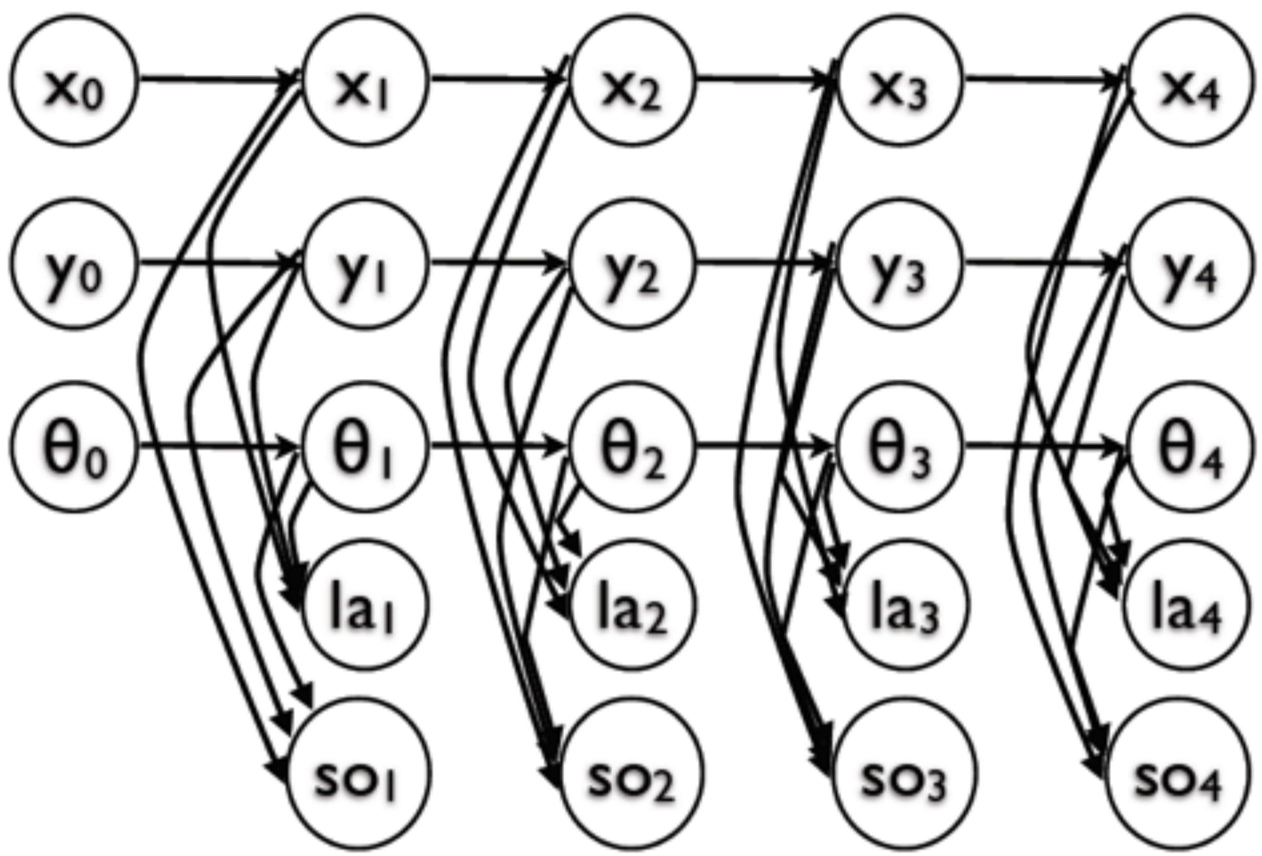
- Hidden Markov Models are Bayes Nets with a polytree structure
- Variable elimination is
  - Linear with respect to number of time slices
  - Linear with respect to largest CPT ($P(s_t|s_{t-1})$ or $P(o_t|s_t)$)

Dynamic Bayes Nets

- What if the number of states or observations are exponential?
- Dynamic Bayes Nets
  - **Idea**: Encode states and observations with several random variables
  - **Advantage**: Exploit conditional independence and save time and space
  - **Note**: HMMs are just DBNs with one state variable and one observation variable

Example: Robot Localization

- **States**: (x,y) coordinates and heading θ
- **Observations**: laser and sonar readings, la and so



DBN Complexity

- Conditional independence allows us to **represent** the transition and observation models very compactly!
- Time and space complexity of inference: conditional independence rarely helps
  - Inference tends to be exponential in the number of state variables
  - Intuition: All state variables eventually get correlated
  - No better than with HMMs

Non-Stationary Processes

- What if the process is not stationary?
  - **Solution:** Add new state components until dynamics are stationary
  - **Example:** Robot navigation based on (x,y,θ) is nonstationary when velocity varies
    - **Solution:** Add velocity to state description (x,y,v,θ)
    - If velocity varies, then add acceleration,...

Non-Markovian Processes

- What if the process is not Markovian?
  - **Solution**: Add new state components until the dynamics are Markovian
  - **Example**: Robot navigation based on (x,y,θ) is non-Markovian when influenced by battery level
    - **Solution**: Add battery level to state description (x,y,θ,b)

Markovian Stationary Processes

- **Problem**: Adding components to the state description to force a process to be Markovian and stationary may **significantly** increase computational complexity

- **Solution**: Try to find the smallest description that is self-sufficient (i.e. Markovian and stationary)

Summary

- Stochastic Process
  - Stationary
  - Markov assumption
- Hidden Markov Process
  - Prediction
  - Monitoring
  - Hindsight
  - Most likely explanation
- Dynamic Bayes Nets
- What to do if the stationary or Markov assumptions do not hold