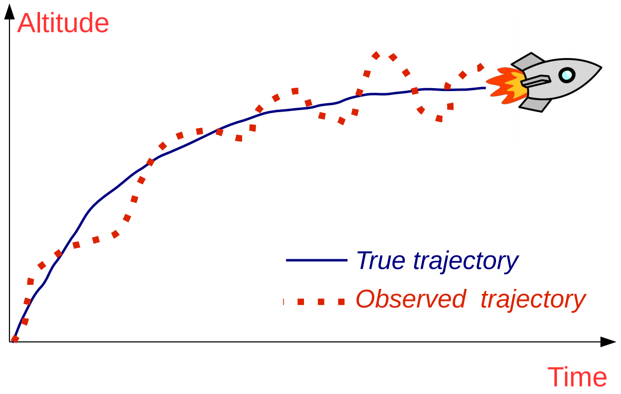
Course 16:198:520: Introduction To Artificial Intelligence
Lecture 12

Kalman Filters, Dynamic Bayesian Networks

Abdeslam Boularias

Monday, November 16, 2015

Objective: identify exactly where the rocket is at the current moment.



Example: Tracking the trajectory of a rocket

Objective: identify exactly where the rocket is at the current moment.

Observation model

- A rocket sends information about its current altitude obtained from sensors, such Inertial Measurement Units (IMU) and satellites.
- These measurements are not precise enough.

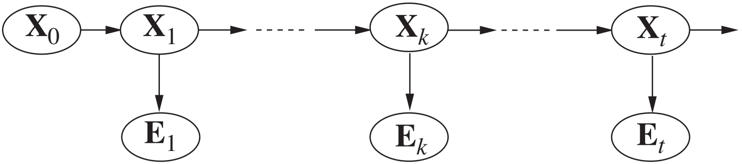
Transition model

- If we know the velocities and accelerations of the rocket, we can predict its current altitude.
- These measurements are also not precise enough (wind, imperfect models).

Problem: use both the observation model and the transition model to estimate the position of the rocket.

Example: Tracking the trajectory of a rocket

Bayesian-network view of the problem



- State X_t is the current position of the rocket. Its true value is unknown.
- Evidence E_t is the estimated current position of the rocket according to the sensors.
- Given all the past and the current observations $e_{1:t} = [e_1, e_2, \dots e_t]$, we want to compute a distribution $P(X_t \mid e_{1:t})$ on the current state.

Filtering (state estimation)

We have seen in the previous lecture how $P(X_t \mid e_{1:t})$ could be calculated recursively by means of the forward procedure:

$$P(X_t \mid e_{1:t}) = \frac{P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}{\sum_{x_t} P(e_t \mid x_t) \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}$$

Problem: In the rocket tracking problem, state X_t corresponds to the altitude, which is a continous variable. There are infinitely many possible values of X_t.

Filtering (state estimation)

We have seen in the previous lecture how $P(X_t \mid e_{1:t})$ could be calculated recursively by means of the forward procedure:

$$P(X_t \mid e_{1:t}) = \frac{P(e_t \mid X_t)\int_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}{\int_{x_t} P(e_t \mid x_t) \int_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}$$

Problem: In the rocket tracking problem, state X_t corresponds to the altitude, which is a continous variable. There are infinitely many possible values of X_t.

Solution: We can derive exactly the same result by replacing sums with integrals.

Filtering (state estimation)

We have seen in the previous lecture how $P(X_t \mid e_{1:t})$ could be calculated recursively by means of the forward procedure:

$$P(X_t \mid e_{1:t}) = \frac{P(e_t \mid X_t)\int_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}{\int_{x_t} P(e_t \mid x_t)\int_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}$$

Problem: How can we calculate

$$\int_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})?$$

Filtering (state estimation)

We have seen in the previous lecture how $P(X_t \mid e_{1:t})$ could be calculated recursively by means of the forward procedure:

$$P(X_t \mid e_{1:t}) = \frac{P(e_t \mid X_t)\int_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}{\int_{x_t} P(e_t \mid x_t)\int_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}$$

Problem: How can we calculate

$$\int_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})?$$

Solution: Assume $P(X_t \mid X_{t-1})$ and $P(E_t \mid X_t)$ are linear Gaussian.

Why do we use Gaussian distributions?

Gaussian distributions are *self-conjugate*: If the prior $P(X)$ is Gaussian and the evidence $P(Y \mid X)$ is Gaussian, then the posterior

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)} = \frac{P(Y \mid X)P(X)}{\int_x P(Y \mid x)P(x)}$$

is also Gaussian.

Kalman Filter

- Independently invented by P. Swerling, T. Thiele and R. Kalman in the 1950's.
- First used for tracking rockets in the Apollo 11 program.
- Widely used for tracking moving objects, stock market data, etc.

If the current distribution $P(x_{t-1}|e_{1:t-1})$ is Gaussian and the transition model $P(X_t|x_{t-1})$ is linear Gaussian, then the one-step predicted distribution given by

$$\begin{aligned} P(X_t \mid e_{1:t-1}) &= \int_{x_{t-1}} P(X_t, x_{t-1} \mid e_{1:t-1}) dx_{t-1} \\ &= \int_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1}) dx_{t-1}, \end{aligned}$$

is also Gaussian.

Kalman Filter

If the current distribution $P(x_{t-1}|e_{1:t-1})$ is Gaussian and the transition model $P(X_t|x_{t-1})$ is linear Gaussian, then the one-step predicted distribution given by

$$P(X_t \mid e_{1:t-1}) = \int_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1}),$$

is also Gaussian.

If the prediction $P(X_t \mid e_{1:t-1})$ is Gaussian and the sensor model $P(e_t|X_t)$ is linear Gaussian, then, after conditioning on the new evidence, the updated distribution

$$P(X_t|e_{1:t}) = \frac{P(e_t|X_t)P(X_t|e_{1:t-1})}{\int_{x_t} P(e_t|x_t)P(x_t|e_{1:t-1})}$$

is also Gaussian.

Kalman Filter

A multivariate Gaussian distribution is uniquely identified by a mean vector μ and covariance matrix Σ.

Reminder

Let $X = (X^1, X^2, \ldots, X^n)$ be a multivariate Gaussian with mean μ and covariance matrix Σ.

- $\mu = \Big(\text{mean of } X^1, \text{mean of } X^2, \ldots, \text{mean of } X^n\Big)$
- $$\Sigma = \begin{bmatrix} \text{Covariance}(X^1, X^1) & \text{Covariance}(X^1, X^2) & \ldots & \text{Covariance}(X^1, X^n) \\ \text{Covariance}(X^2, X^1) & \text{Covariance}(X^2, X^2) & \ldots & \text{Covariance}(X^2, X^n) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Covariance}(X^n, X^1) & \text{Covariance}(X^n, X^2) & \ldots & \text{Covariance}(X^n, X^n) \end{bmatrix}$$

Remember,

$$\text{Covariance}(X_i, X_j) = \int_{x_i}\int_{x_j} (x_i - \text{mean}(x_i))(x_j - \text{mean}(x_j))p(x_i, x_j)dx_idx_j$$

Kalman Filter

The FORWARD operator for Kalman filtering takes a Gaussian forward message $f_{1:t-1}$, specified by a mean μ_{t-1} and covariance matrix Σ_{t-1}, and produces a new Gaussian forward message $f_{1:t}$, specified by a mean μ_t and covariance matrix Σ_t.

$$f_{1:t-1} = (\mu_{t-1}, \Sigma_{t-1}) \xrightarrow{evidence\ e_t} f_{1:t} = (\mu_t, \Sigma_t)$$

Remember

$f_{1:t} = (\mu_t, \Sigma_t)$ defines the state distribution at time t,

$$P(X_t \mid e_{1:t}) = N(\mu_t, \Sigma_t).$$

Linear Gaussian distribution

Let $X \in \mathbb{R}^k$ and $Y \in \mathbb{R}^l$, and F be a $k \times l$ matrix.

$P(X \mid Y)$ is linear-Gaussian if

$$\begin{aligned} P(X \mid Y) &= N(FY, \Sigma) \\ &= \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\Big(-\frac{1}{2}(X - FY)^T \Sigma^{-1} (X - FY) \Big), \end{aligned}$$

where Σ is a covariance matrix, and $|\Sigma|$ is the determinant of Σ.

Note:

P here is a *density function*. The probability distribution is obtained by integrating P within a given integral. The density function is typically denoted by f, we are using a different notation for clarity.

Example: Tracking a Rocket's Altitude

Let's consider a simple example where:

- The state is a single variable $X_t \in \mathbb{R}$ that corresponds to the **true** altitude of a rocket.
- Assume that the velocity of the rocket is constant and equal to v.
- Can we say that $X_{t+1} = X_t + v$?
- What about the wind? what about air friction? what about small accelerations/decelerations that we didn't account for?

Transition (dynamics) model

$$P(X_{t+1} \mid X_t) = N(X_t + v, {\sigma_x}^2) = \alpha \exp\Big(-\frac{1}{2}\frac{(X_{t+1} - (X_t + v))^2}{{\sigma_x}^2}\Big).$$

Rocket Altitude Example

Prior distribution

- We may or may not know the exact initial altitude X_0 of the rocket.
- For example, we are tracking an enemy rocket and we do not know its exact initial altitude when it was first spotted in the sky.
- In general,

$$P(X_0) = N(\mu_0, {\sigma_0}^2) = \alpha \exp\Big(-\frac{1}{2}\frac{(X_0-\mu_0)^2}{{\sigma_0}^2}\Big).$$

Rocket Altitude Example

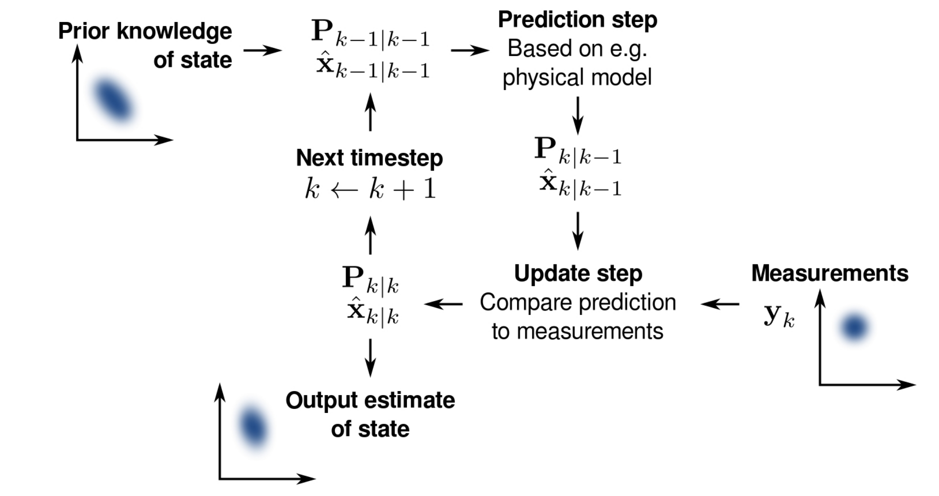
Observation (sensor) model

- We do not have access to the **true** altitude X_t.
- What we observe is a variable $Z_t \in \mathbb{R}$ that corresponds to a **measurement** of the rocket's altitude.
- Can we say that $Z_t = X_t$?
- No, the observation Z_t is a noised estimate of X_t,

$$P(Z_t \mid X_t) = N(X_t, {\sigma_z}^2) = \alpha \exp\Big(- \frac{1}{2} \frac{(Z_t - X_t)^2}{{\sigma_z}^2} \Big).$$

Note: in dynamical systems, we often use Z_t instead of E_t to denote an observation (an evidence).

Kalman Filter Steps



Wikipedia

Rocket Altitude Example

Now, given the prior $P(X_0)$, how can we compute $P(X_1)$?

We need to take into account the following two facts

1. The rocket moved from X_0 to X_1 with velocity v.
2. The measured value of X_1 is Z_1.

Rocket Altitude Example

Now, given the prior $P(X_0)$, the one-step predicted distribution

$$
\begin{aligned}
P(X_1) &= \int_{-\infty}^{\infty} P(X_1 \mid x_0) P(x_0) dx_0 \\
&= \alpha \int_{-\infty}^{\infty} \exp\Big(-\frac{1}{2}\frac{(X_1-(x_0+v))^2}{{\sigma_x}^2}\Big) \exp\Big(-\frac{1}{2}\frac{(x_0-\mu_0)^2}{{\sigma_0}^2}\Big) dx_0 \\
&= \alpha \int_{-\infty}^{\infty} \exp\Big(-\frac{1}{2}\frac{(X_1-(x_0+v))^2}{{\sigma_x}^2} - \frac{1}{2}\frac{(x_0-\mu_0)^2}{{\sigma_0}^2}\Big) dx_0 \\
&= \alpha \int_{-\infty}^{\infty} \exp\Big(-\frac{1}{2}\frac{{\sigma_0}^2(X_1-(x_0+v))^2 + {\sigma_x}^2(x_0-\mu_0)^2}{{\sigma_x}^2{\sigma_0}^2}\Big) dx_0 \\
&= \alpha \exp\Big(-\frac{1}{2}\frac{(X_1-(\mu_0+v))^2}{{\sigma_x}^2+{\sigma_0}^2}\Big) \underbrace{\int_{-\infty}^{\infty} \exp\Big(-\frac{1}{2}\frac{(x_0-a)^2}{b^2}\Big) dx_0}_{constant} \\
&= \alpha' \exp\Big(-\frac{1}{2}\frac{(X_1-(\mu_0+v))^2}{{\sigma_x}^2+{\sigma_0}^2}\Big)
\end{aligned}
$$

Rocket Altitude Example

To complete the update step, we need to condition on the observation at the first time step, namely, Z_1.

$$
\begin{aligned}
P(X_1 \mid Z_1) &= \alpha P(Z_1|X_1)P(X_1) \\
&= \alpha \exp\Big(-\frac{1}{2}\frac{(Z_1 - X_1)^2}{{\sigma_z}^2}\Big)\exp\Big(-\frac{1}{2}\frac{(X_1 - (\mu_0 + v))^2}{{\sigma_x}^2 + {\sigma_0}^2}\Big) \\
&= \alpha \exp\Big(-\frac{1}{2}\frac{(X_1 - \mu_1)^2}{{\sigma_1}^2}\Big) \\
&= N(\mu_1, {\sigma_1}^2).
\end{aligned}
$$

with

$$
\begin{aligned}
\mu_1 &= \frac{({\sigma_0}^2 + {\sigma_x}^2)Z_1 + {\sigma_z}^2(\mu_0 + v)}{{\sigma_0}^2 + {\sigma_x}^2 + {\sigma_z}^2}, \\
{\sigma_1}^2 &= \frac{({\sigma_0}^2 + {\sigma_x}^2){\sigma_z}^2}{{\sigma_0}^2 + {\sigma_x}^2 + {\sigma_z}^2}.
\end{aligned}
$$

Rocket Altitude Example

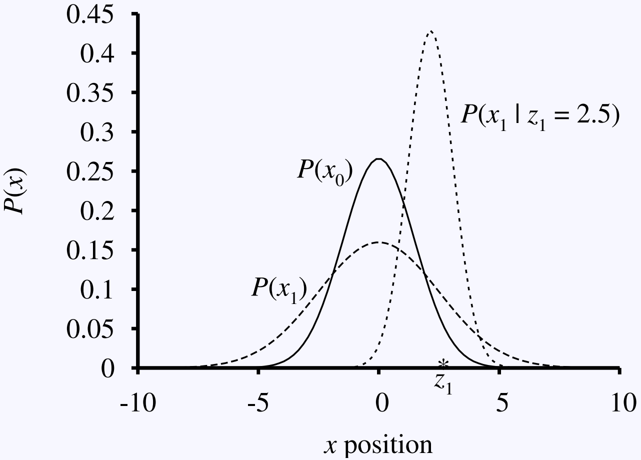
$$P(X_1 \mid Z_1) = N(\mu_1, {\sigma_1}^2).$$

with

$$\begin{aligned} \mu_1 &= \frac{({\sigma_0}^2 + {\sigma_x}^2) Z_1 + {\sigma_z}^2 (\mu_0 + v)}{{\sigma_0}^2 + {\sigma_x}^2 + {\sigma_z}^2}, \\ {\sigma_1}^2 &= \frac{({\sigma_0}^2 + {\sigma_x}^2) {\sigma_z}^2}{{\sigma_0}^2 + {\sigma_x}^2 + {\sigma_z}^2}. \end{aligned}$$

Notice

- $\lim_{\sigma_0, \sigma_x \to 0} \mu_1 = \mu_0 + v$.
- $\lim_{\sigma_z \to \infty} \mu_1 = \mu_0 + v$.
- $\lim_{\sigma_x \to \infty} \mu_1 = Z_1$.
- $\lim_{\sigma_z \to 0} \mu_1 = Z_1$.



Stages in the Kalman filter update cycle: Notice how the variance of $P(X_1)$ is higher than the prior variance of $P(X_0)$, this variance is brought down by the new evidence Z_1.

General Case

Transition (dynamics) model

$$P(X_{t+1} \mid X_t) = N(FX_t, \Sigma_x),$$

where F is a matrix and X_t, X_{t+1} are vectors.

Observation (sensor) model

$$P(Z_t \mid X_t) = N(HX_t, \Sigma_z)$$

where H is a matrix and X_t, Z_t are vectors.

Prior distribution

$$P(X_0) = N(\mu_0, \Sigma_0)$$

General Case

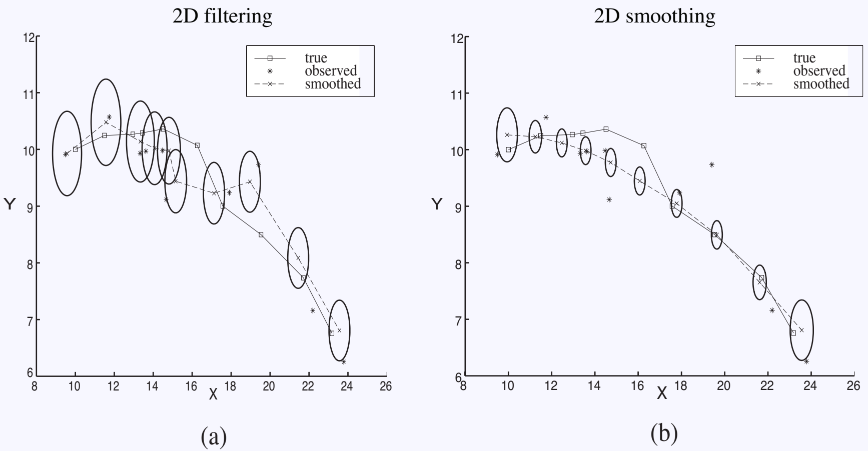
Update equations for the mean and covariance

$$
\begin{aligned}
\mu_{t+1} &= F\mu_t + K_{t+1}(Z_{t+1} - HF\mu_t) \\
\Sigma_{t+1} &= (I - K_{t+1}H)(F\Sigma_t F^T + \Sigma_x),
\end{aligned}
$$

where

$$K_{t+1} = (F\Sigma_t F^T + \Sigma_x)H^T(H(F\Sigma_t F^T + \Sigma_x)H^T + \Sigma_z)^{-1}$$

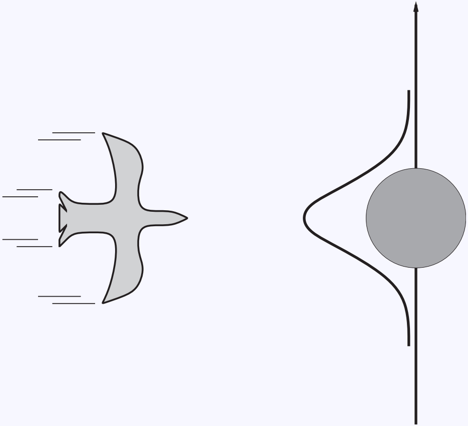
Example



(a) Results of Kalman filtering for an object moving on the X-Y plane, showing the true trajectory (left to right), a series of noisy observations, and the trajectory estimated by Kalman filtering. Variance in the position estimate is indicated by the ovals. (b) The results of Kalman smoothing for the same observation sequence.

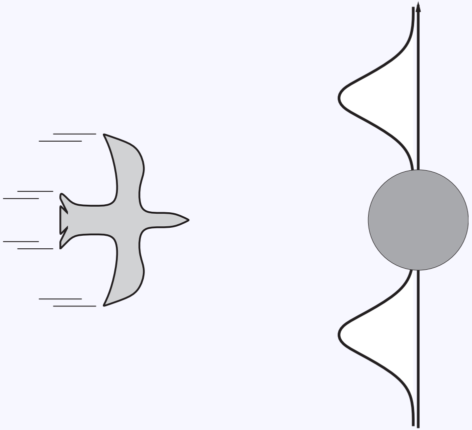
Limitations of Kalman Filters: Multi-modal state distributions

To predict the next position of the bird, we know that the bird will be on the left or on the right of the tree (obstacle).
Using Kalman filters, the predicted state distribution will be a Gaussian.



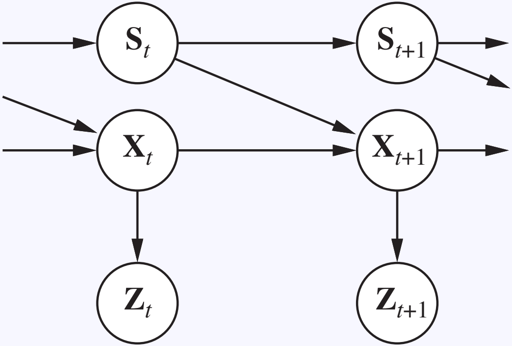
Limitations of Kalman Filters: Multi-modal state distributions

A two-modal Gaussian is more appropriate.



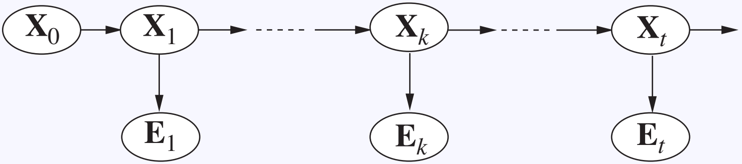
Limitations of Kalman Filters: Multi-modal state distributions

Solution: switching Kalman filter



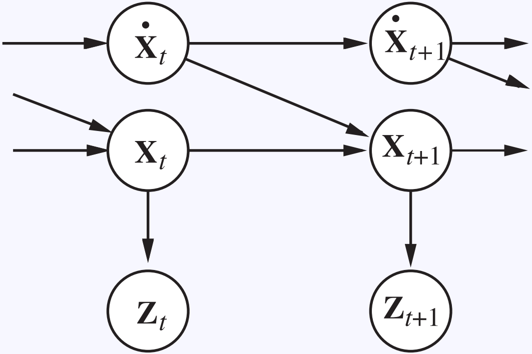
Dynamic Bayesian Networks (DBN)

The temporal models that we saw so far (Markov Chains, Hidden Markov Models, Kalman Filters) are special cases of **Dynamic Bayesian Networks** (DBN).



A Dynamic Bayesian Network is a Bayesian Network where nodes correspond to variables at different times.

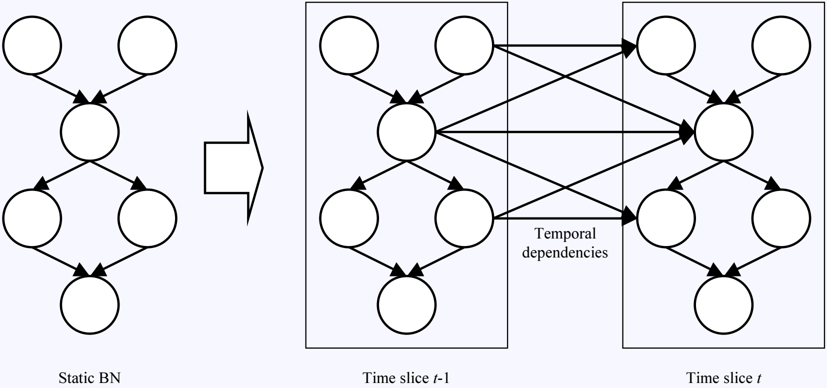
Rocket Altitude Example



Dynamic Bayesian Network for a linear dynamical system with position X_t, velocity $\dot{X}_t$, and position measurement Z_t.

Example of a Dynamic Bayesian Network

A Dynamic Bayesian Network is a Bayesian Network where nodes correspond to variables at different times.



Extending a Bayesian Network to a Dynamic Bayesian Network

from V. Mihajlovic and M. Petkovic. Dynamic Bayesian Networks: A state of the art

Example of a Dynamic Bayesian Networks

Example

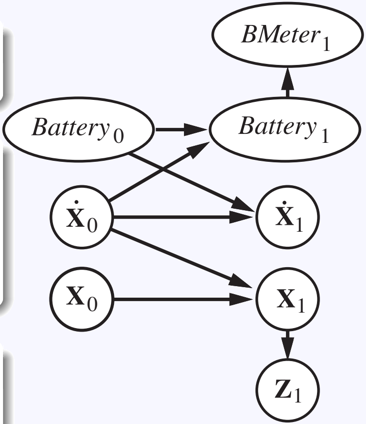
Tracking the position of a mobile robot

Hidden variables

- X_t: true position at time t.
- $\dot{X}_t$: true velocity at time t.
- $Battery_t$: true battery charge at time t.

Observations (evidence)

- Z_t: GPS-provided position at time t.
- $BMeter_t$: measured battery charge at time t.



Two time-slices of
a Dynamic Bayesian Network

Exact Inference in Dynamic Bayesian Networks

- If we **unroll** a Dynamic Bayesian Networks for T time-steps (i.e we graphically represent all the variables from time 1 until T), we obtain a large "ordinary" Bayesian Network.
- Therefore, any inference algorithm for Bayesian Network can also be used for Dynamic Bayesian Networks.
- For example, we can use the variable elimination algorithm.

- A recursive forward procedure can be used to implement the variable elimination algorithm, since we know that a variable at time t cannot be the parent (cause) of anything at time $k < t$.
- Therefore, we don't have to unroll the Dynamic Bayesian Network.

Approximate Inference in Dynamic Bayesian Networks

- We can unroll the Dynamic Bayesian Networks for T time-steps and use likelihood weighting.
- Remember: in likelihood weighting, we generate samples, each sample is an assignment of values to all variables, and we weight each sample by its likelihood.
- Problem: the size of each sample would be (Number of variables in one time-slice) $\times T$.
- Solution: similarly to the recursive forward procedure, generate samples at each time-step and erase the previous ones.

Particles

A particle is a sample: an assignment of values to all the variables in a particular time-step.

Particle Filtering

First, a population of N initial-state samples is created by sampling from the prior distribution $P(X_0)$. Then the update cycle is repeated for each time step:

1. Each sample is propagated forward by sampling the next state value x_{t+1} given the current value x_t for the sample, based on the transition model $P(X_{t+1}|x_t)$.

Particle Filtering

First, a population of N initial-state samples is created by sampling from the prior distribution $P(X_0)$. Then the update cycle is repeated for each time step:

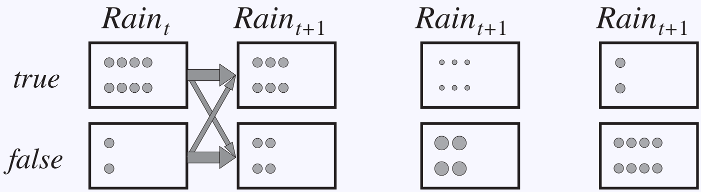
1. Each sample is propagated forward by sampling the next state value x_{t+1} given the current value x_t for the sample, based on the transition model $P(X_{t+1}|x_t)$.
2. Each sample is weighted by the likelihood it assigns to the new evidence, $P(e_{t+1}|x_{t+1})$.

Particle Filtering

First, a population of N initial-state samples is created by sampling from the prior distribution $P(X_0)$. Then the update cycle is repeated for each time step:

1. Each sample is propagated forward by sampling the next state value x_{t+1} given the current value x_t for the sample, based on the transition model $P(X_{t+1}|x_t)$.
2. Each sample is weighted by the likelihood it assigns to the new evidence, $P(e_{t+1}|x_{t+1})$.
3. The population is resampled to generate a new population of N samples. Each new sample is selected from the current population; the probability that a particular sample is selected is proportional to its weight. The new samples are unweighted.

Particle Filtering



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

The particle filtering update cycle for the umbrella DBN with N $=$10, showing the sample populations of each state.
Each sample is weighted by its likelihood for the observation, as indicated by the size of the circles.