# CS 547: Sensing and Planning in Robotics

## Lecture 3: Bayes and Kalman Filters

Gaurav S. Sukhatme
gaurav@usc.edu
robotics.usc.edu/~gaurav

Robotic Embedded Systems Laboratory
Robotics Research Laboratory
University of Southern California
Los Angeles, CA 90089-0781

# Today

- Bayes filter: derivation
- The Markov assumption
- The Kalman filter as a Bayes filter
- Representation

# The Bayes Filter

- We want to estimate state $s$ from data $d$
- $s$ could be robot location, map information, location of targets, *etc.*
- $d$ could be sensor readings *e.g.* range, or actions *e.g.* odometry from encoders
- The Bayes filter **recursively** computes the posterior distribution

$$Bel(s^{(T)}) = P(s^{(T)}|d^{(0\ldots T)})$$

# The Bayes Filter

- Input: $Bel(s), d$
- Output: $Bel'(s)$

if $d = o$ then for all $s$ do

$$Bel'(s) = \eta P(o|s) Bel(s)$$

else if $d = a$ then for all $s$ do

$$Bel'(s) = \int P(s|s', a) Bel(s') ds'$$

## Bayes Filter: derivation

- Key assumption: The world is Markov

$$P(d^{t+1}, d^{t+2}, d^{t+2}, \ldots | s^{(t)}, d^{t}, d^{t-1}, \ldots, d^{0}) =$$
$$P(d^{t+1}, d^{t+2}, d^{t+2}, \ldots | s^{(t)})$$

$$P(d^{t}, d^{t-1}, \ldots, d^{0} | s^{(t)}, d^{t+1}, d^{t+2}, d^{t+2}, \ldots) = P(d^{t}, d^{t-1}, \ldots, d^{0} | s^{(t)})$$

## Bayes Filter: Perception

Case 1: $d^{(T)} = o^{(T)}$

$$P(s^{(T)}|d^{(0\ldots T)}) = P(s^{(T)}|d^{(0\ldots T-1)}, o^{(T)})$$

$$Bel'(s) = \frac{P(o^{(T)}|s^{(T)}, d^{(0\ldots T-1)})P(s^{(T)}|d^{(0\ldots T-1)})}{P(o^{(T)}|d^{(0\ldots T-1)})}$$

$$= \eta P(o^{(T)}|s^{(T)})P(s^{(T)}|d^{(0\ldots T-1)})$$

$$Bel'(s) = \eta P(o|s)Bel(s)$$

## Bayes Filter: Action

Case 2: Let $d^{(T)} = a^{(T)}$

$$P(s^{(T)}|d^{(0\ldots T)}) = P(s^{(T)}|d^{(0\ldots T-1)}, a^{(T)})$$

$$Bel'(s) = \int P(s^{(T)}|d^{(0\ldots T-1)}, a^{(T)}, s^{(T-1)})P(s^{(T-1)}|d^{(0\ldots T-1)}, a^{(T)})ds^{(T-1)}$$

$$= \int P(s^{(T)}|a^{(T)}, s^{(T-1)})P(s^{(T-1)}|d^{(0\ldots T-1)})ds^{(T-1)}$$

$$Bel'(s) = \int P(s|a, s')Bel(s')ds'$$

## Bayes Filter: Implementation

To implement a Bayes filter we need:

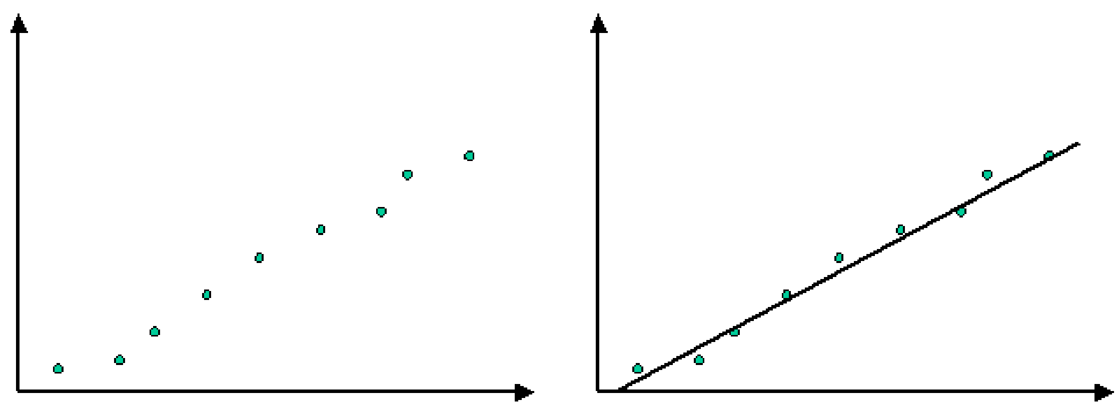
- The update equations (which we just derived)
- An initial belief $P(s^{(0)})$
- A perception model $P(o|s)$
- An action model $P(s|a, s')$
- A representation for the belief function $Bel(s)$

## Bayes Filter: Representation

- Usually the state space $s$ is infinite
- Need some approximations to represent the belief function
- Pick representation based on
  - Computational efficiency
  - Approximation quality
  - Implementation ease

## Parametric Representation

Parametric representations 'compactly' represent data



$(x_1, y_1), (x_2, y_2), (x_3, y_3) \ldots (x_n, y_n)$ $\qquad$ $y = mx + c$

## The Kalman filter as a Bayes Filter

- Kalman filters (KF) represent the posterior belief by a Gaussian (normal) distribution
- A 1D Gaussian is given by

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

- A Gaussian in $d$ dimensions is given by

$$P(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

- A KF is thus a particular parametric representation of a belief posterior

## The Kalman filter is a Bayes Filter

... with the following assumptions:

- Initial belief $Bel(s^{(0)})$ is normal
- State at $t+1$ is a linear function of state at $t$

$$s^{(t+1)} = As^{(t)} + Ba^{(t)} + \epsilon_{action}^{(t)}$$

- Observations are linear in the state

$$o^{(t)} = Cs^{(t)} + \epsilon_{observation}^{(t)}$$

- $\epsilon_{action}^{(t)}$ and $\epsilon_{action}^{(t)}$ are both zero-mean random variables which are normally distributed with covariance $\Sigma_{action}$ and $\Sigma_{observation}$

These assumptions guarantee that the posterior belief is a Gaussian, and there is a very efficient algorithm to compute the posterior called the Kalman Filter.

# The Kalman filter

- $P(s|a, s')$ is a Gaussian whose mean is linear in $a$ and $s$ and whose covariance is $\Sigma_{action}$
- $P(o|s)$ is a Gaussian whose mean is linear in $s$ and whose covariance is $\Sigma_{observation}$
- Thus each belief function uniquely characterized by its mean $\mu$ and covariance matrix $\Sigma$
- Computing the posterior amounts to updating the new $\mu$ and $\Sigma$ from the old using either action or observation
- Key limitation: unimodality

## The Kalman filter Algorithm

- Input: $Bel(s) = (\mu, \Sigma), d$
- Output: $Bel'(s) = (\mu', \Sigma')$

if $d = o$ then

$$\begin{aligned} K &= \Sigma C^T (C \Sigma C^T + \Sigma_{observation})^{-1} \\ \mu' &= \mu + K(o - C\mu) \\ \Sigma' &= (I - KC)\Sigma \end{aligned}$$

else if $d = a$ then

$$\begin{aligned} \mu' &= A\mu + Ba \\ \Sigma' &= A \Sigma A^T + \Sigma_{action} \end{aligned}$$

return $Bel'(s) = (\mu', \Sigma')$

## Piecewise Constant Representations

- Represent belief posterior by piecewise constant functions
- Break state space into contiguous regions
- Assume belief density is uniform within a region
- Tradeoff accuracy and computational efficiency

- Advantages: multimodal
- Disadvantages: Computationally complex

## Static and Dynamic Piecewise Constant Representations

- Static: single spatial decomposition, chosen in advance
  - Topological: coarse decomposition depending on the structure of the environment
  - Metric: equally spaced grid, higher resolution than topological representation
- Dynamic: adapt to shape of posterior
  - Density trees: recursively split environment into regions
  - Selective updating: static decomposition, but update only a subset of cells at a time