CS354

Nathan Sprague

February 19, 2015

State Estimation

- The goal is to estimate the state of the robot from a history of observations:

$$Bel(X_t) = P(X_k \mid Z_1, Z_2, ..., Z_k)$$

- We make some (true-ish) simplifying assumptions:
  - Markov Assumption:

$$P(X_k \mid X_1, X_2, ..., X_{k-1}) = P(X_k \mid X_{k-1})$$

  - Assumption that the current observation only depends on the current state:

$$P(Z_t \mid X_1, Z_1, X_2, ..., Z_{t-1}, X_t) = P(Z_t \mid X_t)$$

Probabilistic State Representations: Grid-Based

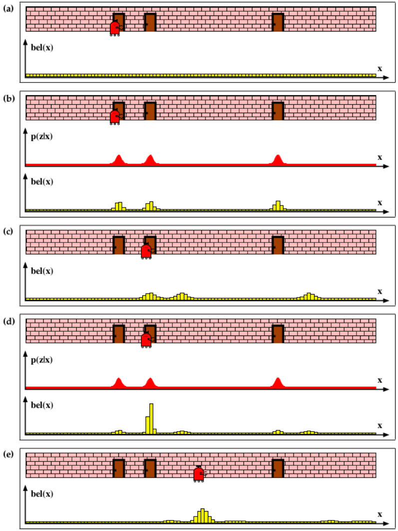

Figure 8.1 Grid localization using a fine-grained metric decomposition. Each picture depicts the position of the robot in the hallway along with its belief $bel(x_t)$, represented by a histogram over a grid.

Probabilistic Robotics. Thrun, Burgard, Fox, 2005

Probabilistic State Representations: Continuous

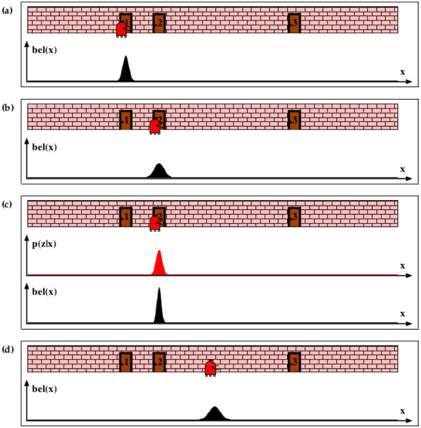

Figure 7.6 Application of the Kalman filter algorithm to mobile robot localization. All densities are represented by unimodal Gaussians.

Probabilistic Robotics. Thrun, Burgard, Fox, 2005

The Answer! Recursive State Estimation

- Two Steps:
  - Prediction based on system dynamics:

$$Bel^{-}(X_t) = \sum_{x_{t-1} \in X} P(X_t \mid x_{t-1}) Bel(x_{t-1})$$

  - Correction based on sensor reading:

$$Bel(X_t) = \alpha P(Z_t \mid X_t) Bel^{-}(X_t)$$

Repeat forever.
Again α is a normalizing constant chosen to make the distribution sum to 1.

The Answer! Recursive State Estimation (continous)

- Two Steps:
  - Prediction based on system dynamics:

$$Bel^{-}(x_t) = \int p(x_t \mid x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

  - Correction based on sensor reading:

$$Bel(x_t) = \alpha p(z_t \mid x_t) Bel^{-}(x_t)$$

Repeat forever.
Again α is a normalizing constant chosen to make the distribution sum to 1.

Prediction Example

- The robot is now moving Right! (or trying to)
- 80% chance to move right, 20% chance stays in place.
- Assume we know that the robot starts in position a,
  $Bel(X_0) =$

a	b	c	d
1	0	0	0

- Or:
  $Bel(X_0 = a) = 1$
  $Bel(X_0 = b) = 0$
  ...

Prediction Example

- Run one step of prediction:

$$Bel^{-}(X_1 = a) = \sum_{x_0 \in X} P(x_1 = a \mid x_0) Bel(x_0)$$

$$= P(X_1 = a \mid X_0 = a) Bel(X_0 = a) +$$
$$P(X_1 = a \mid X_0 = b) Bel(X_0 = b) +$$
$$P(X_1 = a \mid X_0 = c) Bel(X_0 = c) +$$
$$P(X_1 = a \mid X_0 = d) Bel(X_0 = d)$$

$$= .2 \times 1 + 0 \times 0 + 0 \times 0 + .8 \times 0$$

$$= .2$$

Prediction Example

- Similarly
  $Bel^{-}(X_1 = b) = .8 \times 1 + .2 \times 0 + 0 \times 0 + 0 \times 0 = .8$
  $Bel^{-}(X_1 = c) = 0$
  $Bel^{-}(X_1 = d) = 0$
- Unsurprisingly, $Bel^{-}(X_1) =$

a	b	c	d
.2	.8	0	0

Estimation

Now that we have a prediction, we can update it based on the latest sensor reading:

$$Bel(X_t) = \alpha P(Z_t \mid X_t) Bel^{-}(X_t)$$

This is *exactly* what we did when we talked about using Bayes rule to update a prior state estimate based on a sensor reading.

The process is then repeated indefinitely.