Statistical Techniques in Robotics (16-831, F14) Lecture#24 (11/25/2014)

SLAM, FastSLAM and Rao-Blackwellization

Lecturer: Drew Bagnell *Scribe:Venkataramanan Rajagopalan* [1]

1 SLAM

SLAM stands for Simultaneous Localization And Mapping. In the context of (mobile) robots, this scenario is the case where we have no map and no (global) pose and we want both. Two common variants of this problem area solved depending on the results that are sought out. In both cases, we have a set of observations or measurements.

- Batch (or) Smoothing: In this scenario, the goal is to recover the entire pose history and the (best) map
- Online (or) Filtering: In this scenario, the goal is to recover the (best) current pose and the (best) map

In general, solutions to this problem are subject to the "curse of dimensionality" (almost always dominated by the dimensionality of the map)

1.1 Classical Solution to SLAM

The classical approach to solving the SLAM problem with landmarks is the use of X-Kalman Filter. (X - Extended, Unscented).

In a 2D world with a rotating, translating robot, with n landmark (locations), the motion model can be given by the following equation.

$$\mathbf{x_{t+1}} = \begin{bmatrix} x^{t+1} \\ y^{t+1} \\ \theta^{t+1} \\ l_{1,x}^{t+1} \\ l_{1,y}^{t+1} \\ l_{2,x}^{t+1} \\ l_{2,y}^{t+1} \\ . \\ . \\ . \\ l_{n,x}^{t+1} \\ l_{n,y}^{t+1} \end{bmatrix} = \begin{bmatrix} x^{t} \\ y^{t} \\ \theta^{t} \\ l_{1,x}^{t} \\ l_{1,y}^{t} \\ l_{2,x}^{t} \\ l_{2,y}^{t+1} \\ . \\ . \\ . \\ l_{n,x}^{t} \\ l_{n,y}^{t} \end{bmatrix} * \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} + \epsilon \tag{1}$$

Assuming that we get range and bearing measurements for landmarks (with known correspondence),

the observation model for one such range and bearing can be written as

$$\mathbf{z_{t+1}} = \begin{bmatrix} r^{t+1} \\ b^{t+1} \end{bmatrix} = \begin{bmatrix} \sqrt{(x^{t+1} - l_{1,x}^{t+1})^2 + (y^{t+1} - l_{1,y}^{t+1})^2} \\ atan2((y^{t+1} - l_{1,y}^{t+1})^2, (x^{t+1} - l_{1,x}^{t+1})^2) \end{bmatrix} \tag{2}$$

- Running an X-KF in the above setup typically is typically cubic in the dimension of landmarks
- We linearize the observation model about the current range and bearing measurements.

1.2 Typical problems, (hacky) solutions

- Initialization is somewhat of a black art. Typically, it is done by taking multiple observations, assuming that odometry is good.
- Given that the robot's pose is unknown relative to each other, the location of the landmarks get corrleated
- Data association is hard
  - We can associate the observation to the landmark by checking the kalman innovation - predict the location of the landmark and check the observation
  - Evaluate the probability of observation of each of the landmarks. If any of them are low, spawn a new landmark
- Large maps are handled by factorizing the giant matrix into smaller sub-matrices and updating them

2 Rao-Blackwellization

Particle filtering in high dimensional state-spaces can be inefficient because a large number of samples are necessary to represent the posterior distribution.
A standard technique to increase the efficiency of sampling techniques is to reduce the size of the state space by marginalizing out some of the variables analytically - this is called Rao-Blackwellization [1].
The combination of these two approaches is called Rao-Blackwell Particle Filter (RBPF).

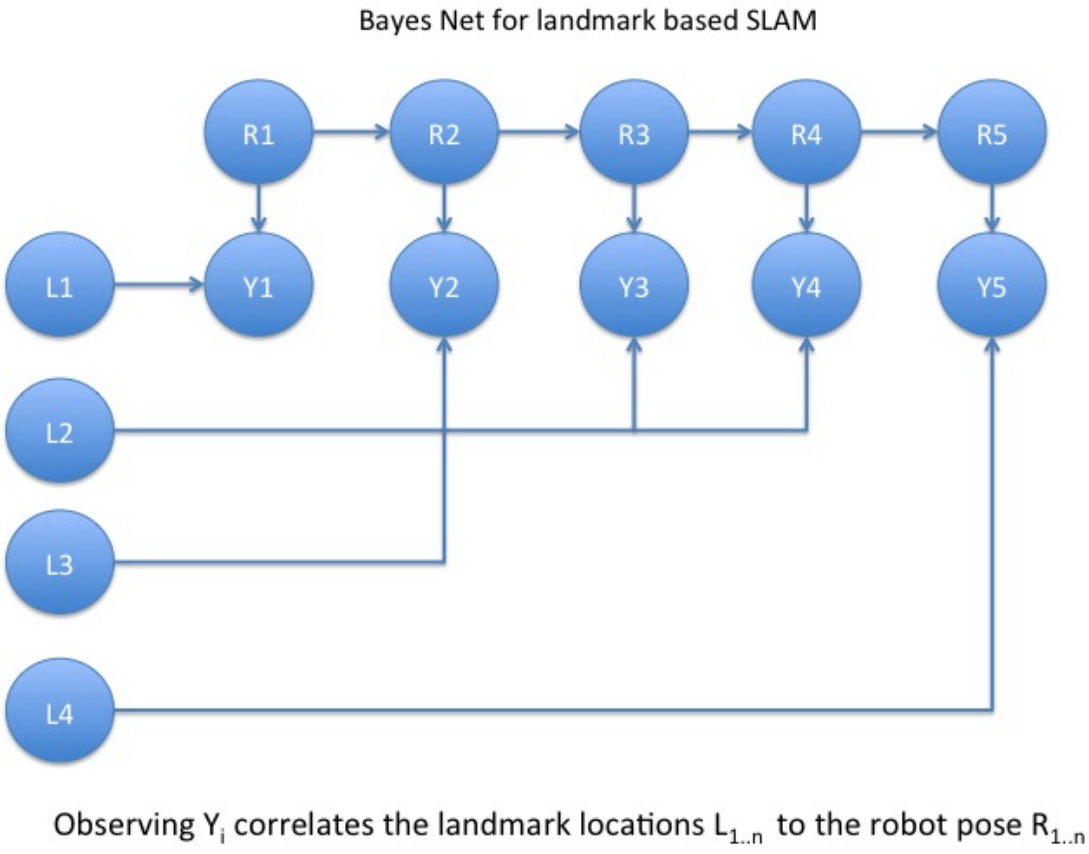
2.1 FastSLAM 1.0

In bayesian filtering, the goal is to compute the following (posterior) distribution $P(Z_t \mid y_{1:t})$ where Z_t is the hidden state at time t and $y_{1:t}$ is the history of all the observations upto time t. Suppose that we can partition the state-space Z_t into two sub-spaces R_t and X_t. by the chain rule of probability, we can write

$$P(X_{1:t}, R_{1:t} \mid y_{1:t}) = P(X_{1:t} \mid R_{1:t}, y_{1:t}) * P(R_{1:t} \mid y_{1:t}) \tag{3}$$

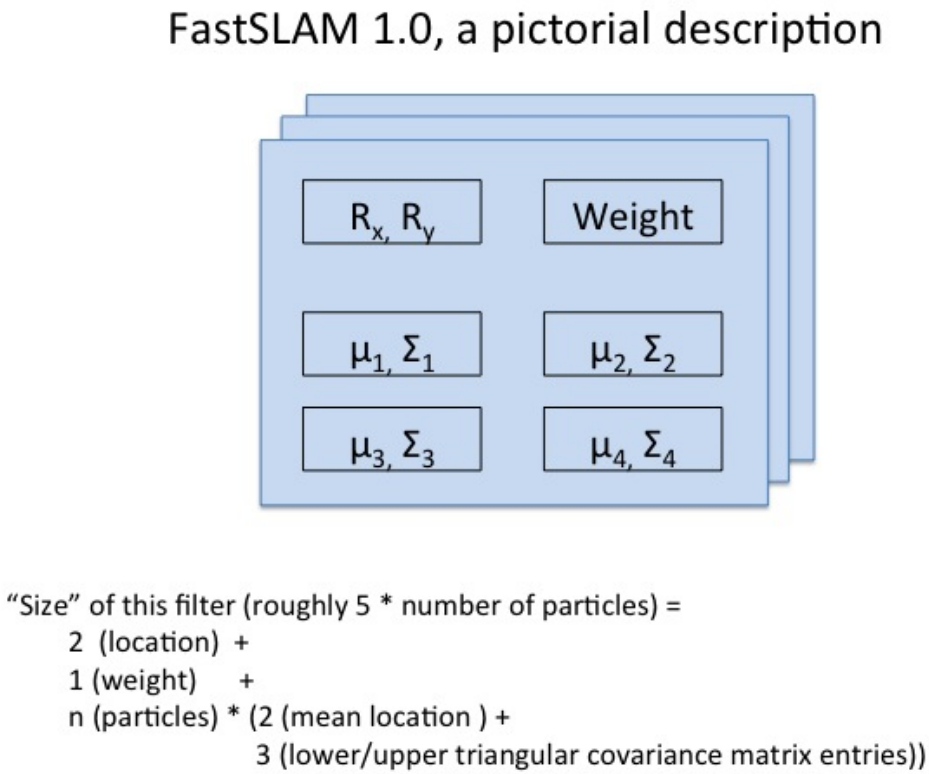
If we can update $P(X_{1:t} \mid R_{1:t}, y_{1:t})$ analytically and efficiently, we only need to sample $P(R_{1:t} \mid y_{1:t})$ using the particle filter. This is the foundation upon which FastSLAM 1.0 [2] is built.

The following figure illustrates the generative model that underlies the SLAM literature.



It is evident from the diagram that the SLAM problem exhibits important conditional independences. In particular, knowledge of the robot's path history renders the individual landmark measurements independent. Thus, the problem of estimating landmarks (collectively) can be decoupled into a collection of estimating the location of individual landmark independently.

FastSLAM 1.0 uses a modified particle filter for estimating the posterior over robot paths. Each particle possesses **K** Kalman filter that estimate the **K** landmark locations condition on the path estimate. A naive implementation of this idea leads to an algorithm that requires $O(\mathbf{MK})$ time where **M** is the number of particles and **K** is the number of landmarks. FastSLAM uses a tree structure to reduce the algorithmic requirements to $O(\mathbf{M}\ \mathbf{log}\mathbf{K})$



2.1.1 Motion Model

Here, we completely ignore the landmark filters and directly apply the motion model to the robot location

2.1.2 Observation Model

Given the new observation and a known-data association,

- Update the Kalman filter **k** for landmark **k** for particle **i**
- Update the weight of the particle **i** as

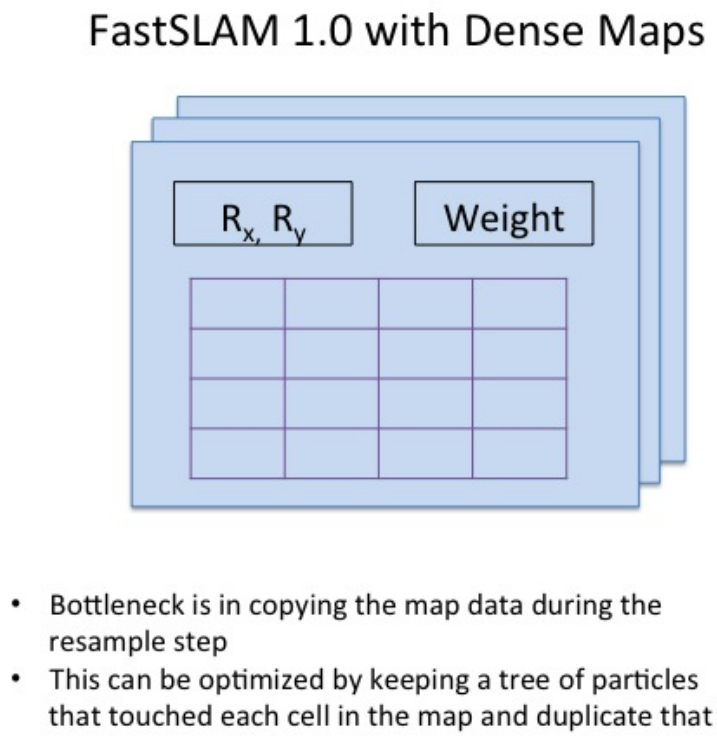
$$Weight_{\mathbf{i}} \leftarrow \frac{\mathbf{1}}{\mathbf{Z}} \times exp((\mathbf{y} - \mu_{\mathbf{y}})^T \sum_{yy}^{-1} (\mathbf{y} - \mu_{\mathbf{y}})) \tag{4}$$

Where,
μ_y is the mean landmark location estimated by the kalman filter
$\sum_{yy}^{-1}$ is the current covariance plus the sensor's noise covariance

2.2 Why should this algorithm work?

We can imagine a case where all the landmarks become uncorrelated by retaining the entire path of the robot. It is straight forward to see that by just keeping the current state estimate, by markovian assumption, this algorithm should work (hand wavy - we should derive this)

2.3 FastSLAM on Dense/Occupation Grid type Maps



References

[1] Neil Gordon Arnaud Doucet, Nando de Freitas. *Sequential Monte Carlo Methods in Practice.* Springer, 2005.

[2] Daphne Koeller Ben Wegbriet Michael Montemerlo, Sebastian Thrun. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem.*