Recursive EM-SLAM Derivation

Chaojie Feng

February 27, 2019

1 Application of EM Algorithm on SLAM

1.1 Data Specification

One of the trick part in application of EM algorithm is to determine what data is complete and what data is incomplete. According to Cappe and Moulins [1], as well as original EM publication from Dempster et al. [2], choice of complete data and incomplete data is usually left to researcher by intuition, and in general by viewing EM algorithm as latent model, complete data can consist of censored observations, noise observations and missing data. In SLAM problem, one intuitive configuration might be specifying observation O as incomplete data, and robot's states S as complete data. This might not be a good choice if we view the EM algorithm from latent model point of view, because our observation O can be viewed as noise observations and there exist many-to-one mapping on O to S. Thus, a better choice would be specifying both observations and states as complete data, $X = \{O, S\}$, and specifying observations also as incomplete data $Y = \{O\}$

1.2 Batch EM Algorithm on SLAM

Assume at instant t, $\phi = [\phi_1, \phi_2, ..., \phi_m]$ are m landmarks in map, $s_t = [p_s, p_y, \theta]$ is robot's state. $o_t = [o_1, o_2, ..., o_m]$ are observations on each landmarks. k denotes currently the algorithm is at k^{th} iteration. s_t and o_t consists complete data denoted as X_t, o_t is also included in incomplete data, denoted as Y_t. In E-step, we can construct Q function as

$$Q(\phi, \phi^k) = \mathsf{E}\{\log [p(s_1, ..., s_t, o_1, ..., o_t)|\phi^k]|o_1, ..., o_t\}$$

Which can be simplified using complete data and incomplete data notation as:

$$Q(\phi, \phi^k) = \mathsf{E}\{\log [p(X)|\phi^k]|Y\}$$

where

$$p(X|\phi^k) = \prod_{i=1}^{t} p(o_t|s_t, \phi^k) p(s_t|s_{t-1}, \phi^k) p(s_0)$$

$p(s_0)$ denotes some prior on robot's states Q function can be numerically approximated using E-RTS smoother. In M-step, we can simply find $\phi^{k+1} = \arg\max Q(\phi, \phi^k)$ using L-BFGS update. The result after numerical approximation is:

$$Q(\phi, \phi^k) = const. - \frac{1}{2} \sum_t ||o_t - h_t(\hat{s_{t|K}}, \phi)||^2 + \mathsf{tr}(F)$$

where F represents some numerical term occurred in E-RTS step.

2 Derivation of Recursive EM-SLAM algorithm

2.1 Stochastic Gradient EM algorithm on SLAM

Note that in batch EM approach, we need lots of iterations and makes batch update on robot's complete data, as t increases, computational cost also increases, which makes this approach not difficult to implement in a real system. A sequential algorithm is derived based on stochastic gradient EM algorithm, followed by a further modified online version. In this section, we will abandon all iteration index k, and the number of iteration on EM is the same as number of observations the robot has taken, denoted as n. Titterington [3] has proposed first online version of EM algorithm for parameter estimation, and in SLAM setting, the algorithm can be described as:

$$\phi_{n+1} = \phi_n + \gamma_{n+1} I^{-1}(\phi_n)\, \mathsf{E}[\nabla_\phi \log p(X_{n+1}|\phi_n)|Y_{n+1}]$$

Where I represent Fisher information matrix, and γ_{n+1} denotes a deccaying step size. This equation is in analogy with equation (8) of section 2.2 in Cappe and Moulins' derivation [1]. It is also in accordance with the formulation on another recursive EM-SLAM on Multi-Target Tracking proposed by Frenkel and Feder [3]. Notice that there are two difficulties here: (1) it is difficult to evaluate the inverse of Fisher information matrix. (2) it is difficult to calculate the gradient.

2.2 Re-formulation based on Stochastic Approximation

Due to computational difficulties in traditional formulation in 2.1, here is a better modification. Consider the following equation:

$$\hat{Q}_{n+1}(\phi) = \hat{Q}_n(\phi) + \gamma_{n+1}(\mathsf{E}[\log p(X_{n+1}|\phi)|Y_{n+1}] - \hat{Q}_n(\phi))$$

This equation is equivalent to previous formulation in section 2.1 but rather abandoned difficulties in calculating Fisher information matrix and gradients. This equation is also subjected to some assumptions and it is of practical interest only if we can compute $\hat{Q}_n(\phi)$ efficiently. It is also assumed that the complete data distribution belongs to an exponential family. Luckily, $\hat{Q}_n(\phi)$ can be computed efficiently by numerically estimation step and E-RTS smoothing process in section 1.2, and robot's observations is of Gaussian distribution, which belongs to an exponential family. To be more specific, $\mathsf{E}[\log p(X_{n+1}|\phi)|Y_{n+1}]$ can be thought of as a one-step version of formulation in section 1.2, which will reduce computation largely. Where in section 1.2, Q function is formulated in a batch manner by taking expectation of whole batch of available data. In the new formulation, the expectation step is simply taking the most recent data X_{n+1} and Y_{n+1} to formulate $\hat{Q}_{n+1}(\phi)$ due to we have already calculated Q function $\hat{Q}_n(\phi)$ when robot took observation in a previous step.

3 Reference

[1] Capp, Olivier, and Eric Moulines. "Online expectationmaximization algorithm for latent data models." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71.3 (2009): 593-613.

[2] Titterington, D. Michael. "Recursive parameter estimation using incomplete data." Journal of the Royal Statistical Society: Series B (Methodological) 46.2 (1984): 257-267.

[3] Frenkel, Liron, and Meir Feder. "Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking." IEEE Transactions on Signal Processing 47.2 (1999): 306-320.