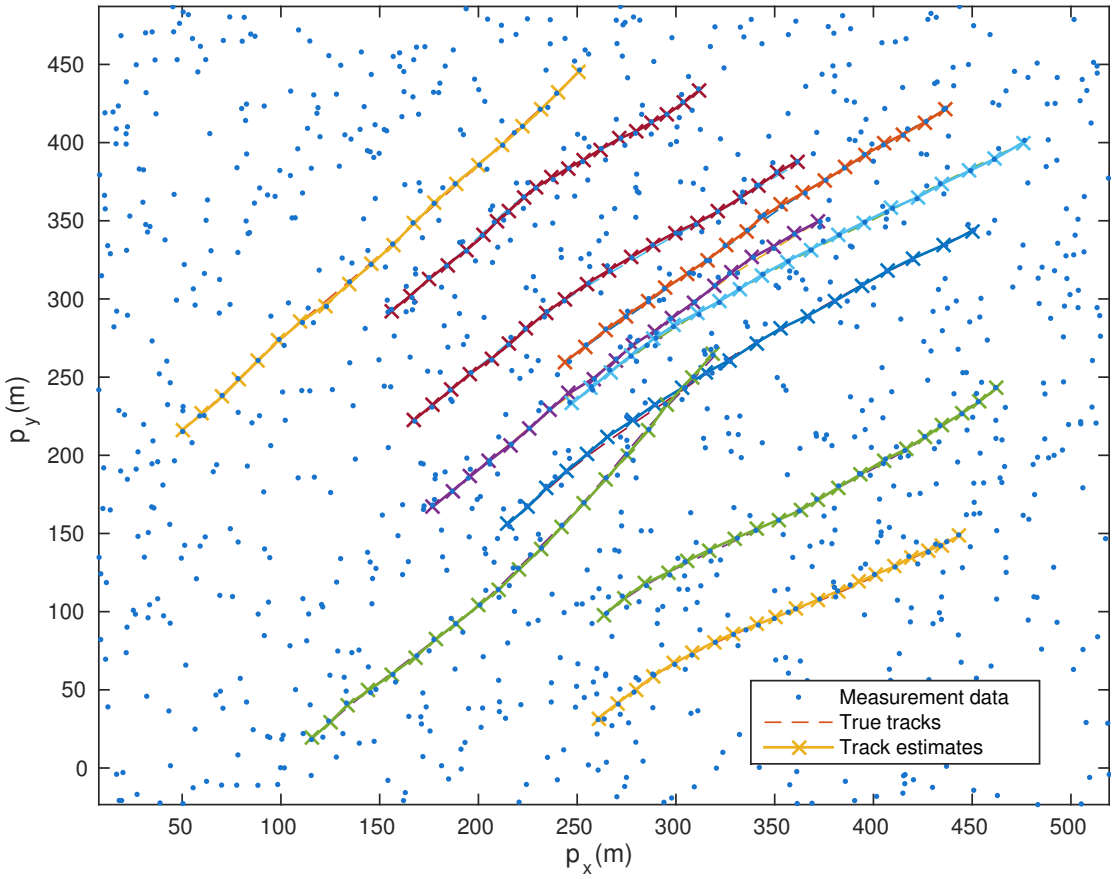
CHALMERS



Bayesian tracking of multiple point targets using expectation maximization

Master's thesis in Communication Engineering

RAGHAVENDRA SELVAN

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

MASTER'S THESIS IN COMMUNICATION ENGINEERING

Bayesian tracking of multiple point targets using expectation maximization

by

RAGHAVENDRA SELVAN

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

Bayesian tracking of multiple point targets using expectation maximization
RAGHAVENDRA SELVAN

Master's thesis EX037/2015
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Illustration of a typical point target tracking scenario, estimated using one of the proposed EM-based algorithms.

Chalmers Reproservice
Göteborg, Sweden 2015

Bayesian tracking of multiple point targets using expectation maximization
Master's thesis in Communication Engineering
RAGHAVENDRA SELVAN
Department of Signals and Systems
Chalmers University of Technology

Abstract

The range of applications where target tracking is useful has grown well beyond the classical military and radar-based tracking applications. With the increasing enthusiasm in autonomous solutions for vehicular and robotics navigation, much of the maneuverability can be provided based on solutions that can track multiple targets, in a computationally inexpensive and accurate manner.

This thesis is concerned with solving the problem of tracking multiple *point* targets, which is a common occurrence in many tracking applications. The main challenge in multi-target tracking is to resolve measurement-to-target association uncertainties, also called the data association uncertainties. Using the Bayesian approach, these uncertainties are modeled as random variables; where, each instance of the data association variable corresponds to a unique data association hypothesis. The objective, then, is to use the measurements from sensors to choose the best data association hypothesis, from which the estimates of target trajectories can be obtained. In an ideal world, we could maintain all possible data association hypotheses from observing all measurements, and pick the best hypothesis. But, it turns out the number of data association hypotheses grows exponentially with the number of measurements over time, rendering this optimal solution intractable. Vast literature in the multi-target tracking is dedicated to solving this problem tractably.

In this thesis, a *variational Bayesian* approach has been used, more specifically, the main contribution is to use *expectation maximization* (EM) in tracking multiple point targets. EM is an iterative algorithm that can be used to approximate the best data association hypotheses, and/or target state estimates in a computationally efficient manner. Depending on, if the overall joint density is maximized over the data association variables, or over the target state variables, two EM-based algorithms for tracking multiple point targets are derived, implemented and evaluated. In the first algorithm, the data association variable is integrated out, and the target states are estimated. In the second algorithm, the data association variable is estimated, while the target states are integrated out. In the end, both the algorithms yield the desired target states. It is shown that the two proposed algorithms can be implemented using existing, simpler solution blocks like Bayesian smoothing, 2-D auction algorithm and computation of marginal data association probabilities. Performance of the two proposed algorithms are compared with existing multi-target algorithms, to highlight the overall improvement in performance, and possible future extensions to the work are presented.

Keywords: multi-target tracking, Bayesian filtering, variational Bayes, expectation maximization

Acknowledgements

In retrospection, it appears that I had not quite anticipated the steep learning curve I would have to scale, when I embarked upon this research-intensive thesis. It now seems to have culminated into a productive piece of work, that I am proud of. This certainly would not have been possible, without the guidance and support of many important people.

First of all, I would like to thank my supervisor, Lennart Svensson. Not only for his supervision pertaining to the thesis, but also for being an impeccable example of how, why and what a researcher, and a teacher should be. You have truly been an inspiration Lennart, and I will try my best to emulate you.

Collaborating can be an enriching experience, more so, when we are on the receiving side. Thank you Abu Sajana for being on other side, and helping me out by entertaining the pettiest of my queries and engaging me in insightful discussions. It has been one of the best professional interactions I have had. Thank you for facilitating a conducive environment for me to learn, and perform. You are a good supervisor. If there was still a doubt.

My gratitude to Maryam for the insightful discussions and also for the comprehensive reviewing of my thesis. Thanks to Malin, Lars, Anders and others in the group for the support in different forms.

Love to my family and my girl friend Sneha, for putting up with my tantrums, during those occasional bad days, and for being ever so supportive in everything I have ever done.

Thanks to Ashfaq, for being such a non-intrusive and supportive room-mate, and more importantly a good friend. Cheers to the thesis room friends for the collective calmness through the ten months of our stay in there, and the diverse discussions during coffee-breaks and the pizza times.

Last, but certainly not the least, a big kudos to the Free Software communities, especially to the Debian community – my platform for years now. And LaTeX: academic writing would not be as tedious, but at the same time, as gratifying without you.

Contents

List of publications

This thesis consists of an extended summary and the following appended papers:

Paper A **Tracking multiple point targets using expectation maximization**

Paper B **Expectation maximization with hidden state variables for multi-target tracking**

Part I

Introductory chapters

Chapter 1

Introduction

For years, the idea of autonomous vehicles and amicable robots has remained an indispensable vision of the future for science fiction writers. While this aspirational vision has been around for decades, it is now, within the last decade, that many such figments from fiction, like self-driving vehicles and nature-emulating robots have actually turned into reality [1], [2]. In the case of autonomous vehicles, for instance, features like adaptive cruise control, forward collision avoidance and blind spot detection have become mainstream, and fully autonomous vehicles have already been tested by many automotive leaders, and has created enormous excitement [3], [4].

Of the many technological advancements, one critical piece that is helping autonomous solutions in many domains evolve is multi-target tracking. Multi-target tracking is the task of simultaneously tracking many targets, through space and time, from observed measurements. Target tracking has long been primarily associated with military and radar-based applications. Of late, driven by the surge in computation power, interest in many diverse applications that use multi-target tracking is on the rise, spanning domains like autonomous navigation, computer vision and surveillance.

In this thesis, we deal with the problem of tracking multiple *point* targets. The point nature of targets arises mainly from the sensor's perspective. Depending on how the sensor perceives targets and generates measurements from them, targets can be broadly categorized into - extended and point targets. When a target has more than one scattering centre, i.e, when a single target can generate multiple measurements, it is then described as an extended target [5], whereas for point targets there is a single scattering centre, meaning, each target is assumed to generate only one point measurement. One other feature of point targets, that makes the problem of tracking them interesting, is that the measurements typically do not carry any target signatures.

The implication of point measurements bearing no target signatures is that, when we receive a bunch of measurements, we have no accurate way of associating measurements to different targets. Further, it is not uncommon for sensors to generate measurements of non-target objects, called fasle detection or clutter. As a consequence, we end up having two kinds of uncertainties pertaining to point target measurements. In the first case, there is uncertainty in distinguishing measurements from the targets of interest, to the ones generated by clutter. In the second case, the ambiguity is in associating measurements between the actual targets. These uncertainties in associating measurements with targets are termed as data association uncertainties, which will feature as the primary challenges we are solving in this thesis.

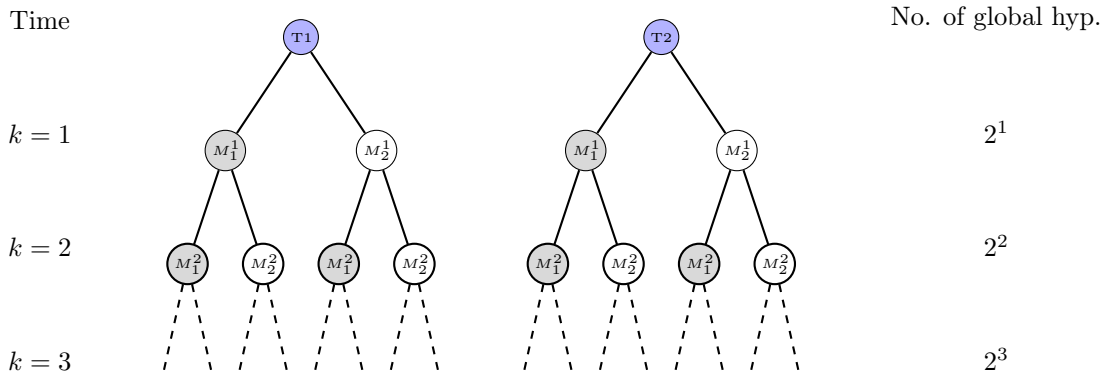


Figure 1.1: *Exponential growth of data association hypotheses, for the case of tracking two targets, with two measurements generated at each time scan.*

Adding further to the problem of data association uncertainties is the chance of missing detections from targets. Measurements can be missed either when targets enter shadow regions, where the sensors cannot detect them, or because of the presence of noise in sensor measurements. Noisy measurements can be due to the quality of sensors, for instance, measurements from a higher quality sensor is expected to yield better measurements. It could also be due to the nature of the environment in which the targets are tracked in. These external factors contributing to the measurement noise are commonly handled by modelling them as a random variable. The most common solution to obtain tracks from noisy measurements is to perform, what is naturally called as, filtering.

Bayesian filtering is the most commonly used filtering strategy, used to reduce the impact of noise from measurements, when incorporating the information from them. It offers a concrete framework to estimate variables of interest, in our case target states, which are indirectly observed via noisy measurements. It is based on the broader Bayesian paradigm which, simply put, is philosophically close to our outlook of life. Usually, when we are uncertain about some event, we make our decisions based on our existing beliefs, and when we do observe some relevant evidence, we try and update our beliefs by incorporating the newly available information from the evidence, making us a tad bit more certain in our updated beliefs. In Bayesian filtering, as applied to tracking, we do something similar – we have a prior belief about target states at the start of tracking, which could be with large uncertainty. When we observe measurements, we use the information from these measurements to update our prior beliefs, which results in updated beliefs (posterior distributions) of the target states for the current time instant with lesser uncertainty. We use this new information to predict the target states for the next time instant, which again increases the uncertainty, and when we observe new measurements for that time instant, we update the target states, and so on. If data associations are known by some means, i.e, when measurements are partitioned to belong to different targets, Baysian filtering can be immediately yields target state estimates. Thus, problem of resolving data association uncertainties forms the crux of solving multi-target tracking.

Data association uncertainties are commonly handled by mathematically modelling them as a discrete random variable. This random variable can take values that would form data association hypotheses. One possible way is to allow each instantiation of the data association variable to be a measurement-to-target assignment matrix. The rows of such a matrix can correspond to targets, columns to time instants and each of the entries correspond to the measurement index assigned to the target at a time instant. For this data association matrix to be valid, the entries in it must satisfy two constraints: Firstly, every target must be assigned to at most one target. Secondly,

every measurement must be assigned to only one target. That is, no measurement can occur more than once in the assignment matrix. Assignments which satisfy these constraints are termed as *global* hypothesis, as the assignments are globally (with respect to all targets) compatible, i.e, there is no conflict to assign a particular measurement to more than one target. In an ideal world, one could develop and maintain all possible data association hypotheses, and pick the best sequence based on some criterion (the posterior probability). Given this best data association sequence, target states can be obtained simply by filtering. But, maintaining all the possible data association hypotheses is intractable, as the number of hypotheses grows exponentially with the number of measurements obtained at each scan. We look at a toy example, to gain some intuition of the problem at hand.

Consider the simplest case of tracking two targets – T_1 and T_2. If we wishfully assume there are no clutter measurements, and that we do not miss any detections from the two targets, we obtain two measurements at every time scan. Let M_1^1 and M_2^1 be the two measurements obtained at the first time scan. As the measurements do not carry target signatures, we can build two hypotheses, $k_1 = \{M_1^1 \to T_1, M_2^1 \to T_2\}$ and $k_2 = \{M_2^1 \to T_1, M_1^1 \to T_2\}$, depicted as the first branch of the hypothesis tree in Figure 1.1. At time scan 2, we get two more measurements, M_1^2 and M_2^2. Now, there are four data association hypotheses, and eight at the third scan, which implies, that the number of global data association hypotheses grows exponentially.In fact, the number of global data association probabilities grows as $O(N_t^M)$, where N is the number of targets and M_t is the average number of measurements at time t. If we continue growing the hypothesis tree even up to 10 time scans, we are left with a massive $2^{10} = 1024$ data association hypotheses, for the simple case of tracking two targets, in the best conditions possible. This is already intractable, and only worsens when we consider tens and hundreds of targets, in cluttered environment with chances of missed detections.

As we have seen by now, resolving data association uncertainties is crucial in obtaining a good multi-target tracking solution. Due to the intractability of maintaining all possible data associations, all existing multi-target tracking algorithms try and circumvent this problem in their own ways.

In this thesis, we propose to use the *variational Bayesian* (VB) approach, wherein, we deploy *expectation maximization* (EM) – an iterative algorithm, using which we can approximate the best possible data association hypotheses and/or target state estimates in a computationally efficient manner. Due to its iterative nature, EM beats the intractability that is innate to the problem, of maintaining all possible data association hypotheses. There already exists an EM based multi-target tracking solutions. One of the popular ones is the *probabilistic multiple hypothesis tracking* (PMHT). But, the target assumptions made by PMHT befits extended targets, although they have been widely used to track point targets. We contrast this assumption, and develop the solutions for point targets. Thus, the main contribution in this thesis, is to develop and present EM-based solutions as an approximate, but close to accurate solution yielding strategy for the problem of tracking multiple point targets.

1.1 Outline of the thesis

The thesis is organized into two parts. In the first part, we present the basic theoretical concepts, leading to the formulation of multi-target tracking problem, and methods used in our contributions. We present many ideas from the literature to fit in the context of our research contribution. In the second part of the thesis, we present our main contributions – two EM-based multiple point target tracking algorithms, in the form of two papers.

The first part of the thesis comprises of six chapters. In Chapter 2, fundamentals of estimation

theory are presented in relation to target tracking, followed by a brief discussion on Bayesian filtering and smoothing in Chapter 3. In Chapter 4, the problem of multi-target tracking is formulated, wherein, the challenge of data association is detailed using existing algorithms. Chapter 5 is dedicated to *multiple hypothesis tracking* (MHT), which serves as the benchmark algorithm for evaluating our proposed algorithms. In Chapter 6, we discuss expectation maximization, and make the connection to variational Bayesian method.

1.2 Contributions and future work

We have presented two multiple-point target tracking algorithms, developed using EM in the two appended papers. Although, both the algorithms are based on EM, they are philosophically different tracking algorithms – in one, the data association variable is integrated out, and in the second, maximization is performed over the data association hypothesis.

Paper A: Tracking multiple point targets using expectation maximization

In the multi-target tracking algorithm derived in Paper A, the data association uncertainties are treated as hidden variables. When EM is used in this context, it yields in a density over the data association uncertainties and MAP estimation of the target states. We show that the density over the data association variables can be handled by the computation of marginal data association probabilities. The algorithm is implemented using Loopy Belief Propagation and RTS smoothing. We evaluate the presented algorithm with related algorithms and show that the proposed solution yields improved performance, even in difficult tracking environments.

Paper B: Iterative GNN based EM in tracking of multiple point targets

In Paper B, the target states are treated as hidden variables. When EM is applied to this setting, we obtain a density over the target states and we are required to perform MAP estimation of data association variables. We derive the algorithm and show that it can be implemented using simpler solutions to the 2-D assignment problems and smoothing. The algorithm is also extended to handle crossing targets. We propose an *ad-hoc*, but efficient strategy to detect and handle the scenarios when targets are crossing. Detailed evaluation of the algorithm is presented for different scenarios.

1.3 Author's contributions

The work presented in this thesis is based on close collaboration with Lennart Svensson and Abu Sajana Rahamthullah, during a period of ten months. The author has taken part in all aspects of the development, implementation and documentation of the research presented in this thesis.

1.4 Future work

We foresee many interesting directions to extend the current work. The proposed solutions have been derived and implemented under some assumptions. Generalizing the proposed solutions to encompass a broader range of problems can be useful. We propose three such ideas below.

Handling target birth/death

Both the EM-based algorithms we propose, have been derived under the assumption that we are tracking a known number of targets. In real tracking applications, this is not the case. Targets appear and disappear randomly. Extending our solutions to incorporate strategies to handle birth and death of targets would be an interesting and useful extension.

Online versions of the algorithm

Another assumption, based on which we have derived the solutions, is that, the algorithm works on batches of measurements. This again might not be the case in many real-tracking applications. We can extend our solutions to be able to incorporate observations as they arrive, thus making them online solutions.

Initialization of EM

Expectation maximization forms a core part of our solution. It is a well studied problem that EM is sensitive to initialization. Tackling this sensitivity, in relation to tracking applications, can be good contribution in future work.

Chapter 2

Estimation theory

Multi-target tracking, at its core, can be interpreted as simultaneously solving the data association and state estimation problems. In this thesis we focus on the problem of trajectory estimation and in this chapter we present the fundamental concepts and ideas of parameter estimation from observed data. The observed data in tracking applications can be measurements from sensors such as radars or cameras. The *frequentist* idea of deterministic parameter estimation is presented using *maximum likelihood* (ML) estimation, and the Bayesian paradigm, which is the approach used in this thesis, will be discussed by presenting the *maximum a posteriori* (MAP) and *minimum mean squared error* (MMSE) estimators. First, a brief discussion on some of the useful distributions, that are ubiquitous in the report, are presented.

2.1 Useful distributions

In this section we present two commonly used distributions - the multivariate Gaussian and the Gaussian mixture model distributions, that have wide applicability in target tracking, and are hence, featured recurrently in this thesis. While there are many differences between the two distributions, we emphasize on the fact that the former is a unimodal distribution, whereas the latter can be used to represent multi-modal distributions.

2.1.1 Multivariate Gaussian distribution

The multivariate Gaussian/normal distribution is the most widely used joint probability density function for continuous variables, with many properties that make its use all the more attractive. Firstly, it can be completely described by its first and second moments, which are the mean and covariance matrices, respectively. Secondly, another important feature of the multivariate Gaussian distribution that makes it useful in Bayesian estimation is that it forms a conjugate pair along with a Gaussian likelihood, i.e, when a Gaussian prior is updated with a Gaussian likelihood, it yields a posterior distribution that is Gaussian [6]. This reduces the complexity of many problems immensely.

The multivariate normal distribution for a D-dimensional random variable X is described by its *probability density function* (PDF), parameterized by the mean vector μ and covariance P,

given as,

$$\mathcal{N}(X;\mu,P) = -\frac{1}{(2\pi)^{D/2}|P|^{1/2}}\exp[(X-\mu)^T P^{-1}(X-\mu)]. \tag{2.1}$$

The mean is, $\mu = \mathbb{E}(X) \in \mathbb{R}^D$ and covariance matrix is, $P = \text{cov}(X) \in \mathbb{R}^{D\times D}$.

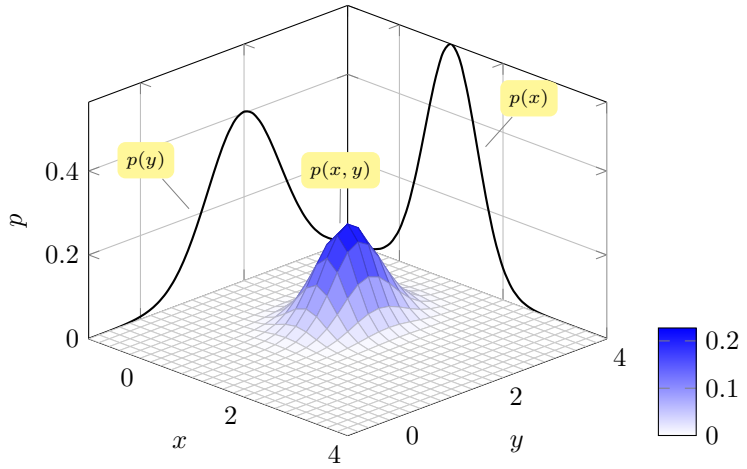


Figure 2.1: *Illustration of a bivariate Gaussian distribution. The marginal and joint probability distributions are denoted, and identified as* $p(x), p(y)$ *and* $p(x, y)$*, respectively.*

In target tracking, the multivariate-Gaussian distribution is used extensively to represent target states, wherein, each of the variables correspond to the target position and velocity. One commonly encountered form of single target state vector comprises of four state variables, $\mathbf{x} = [x, y, v_x, v_y]^T$, where p_x, p_y are the horizontal and vertical positions, and v_x, v_y are the horizontal and vertical velocity components, respectively, and are univariate Gaussian distributions themselves. It is also common to work with likelihood functions in multivariate Gaussian forms, to exploit the conjugate pair feature when performing filtering. Figure 2.1 shows the PDF of a 2-dimensional Gaussian distribution[1], here the variables can be interpreted as the position estimate for a single target. Intuitively, the distribution describes the possible states values which means that the mode of the distribution describes the most probable state value.

2.1.2 Gaussian mixture models

Multivariate Gaussian distribution is a unimodal distribution with a single high probability point. It cannot describe more complex scenarios, where it maybe desirable to have a distribution with multiple modes. For instance, in tracking multiple targets, we get a multi-modal distribution, where the different modes can correspond to different data association hypotheses. In subsequent chapters, we discuss the ramifications of having multi-modal likelihood functions and posterior distributions in multi-target tracking scenarios. In this context, the most commonly used mixture model is the *Gaussian mixture models* (GMM), and it can be seen as an extension of the multivariate Gaussian distributions. A GMM in essence is a weighted sum of Gaussian distributions [7], given as,

$$p(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X;\mu_k,P_k), \tag{2.2}$$

[1]Based on the LaTeXcode from `http://tex.stackexchange.com/a/31715`

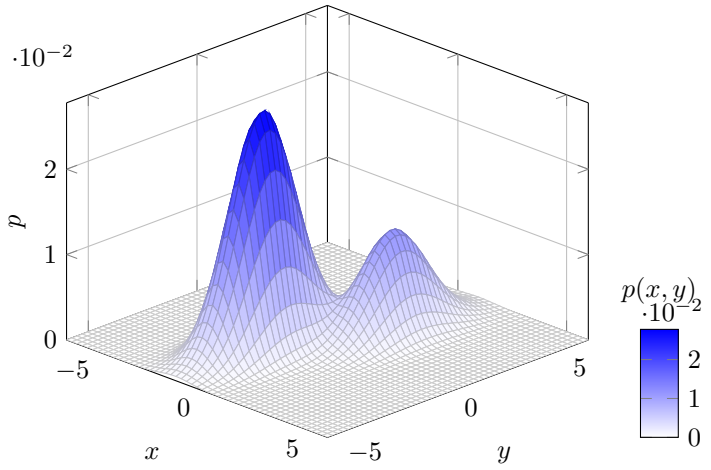


Figure 2.2: *Illustration of a bivariate Gaussian mixture model with two modes.*

where π_k are mixture weights that must sum to one, i.e. $\sum_{k=1}^{K} \pi_k = 1$. A bivariate GMM with two components is shown in Figure 2.2.

2.2 Parameter estimation

Depending on whether the unknown parameters are modelled as deterministic variables or random variables, two schools of parameter estimation methods exist. In the *frequentist* approach, unknown parameters are treated as deterministic variables, that can be estimated from data. In the *Bayesian* approach, unknown parameters are treated as random variables; thus, there is a prior density on them, which is updated using the observed measurements. We look at each of these approaches using popular estimation techniques within their frameworks.

2.2.1 Maximum likelihood estimation

Given a set of observations Y, we are interested in estimating the deterministic, but unknown parameter θ. In *maximum likelihood* (ML) estimation, for the given fixed data, parameters are varied to find the best parameter setting that maximizes the likelihood of the observed data [8]. Thus, ML estimate is the value of θ that maximizes the likelihood function $p(Y|\theta)$, also written as $\ell(\theta|Y)$ to emphasize on the fact that Y is observed. The ML estimation problem is,

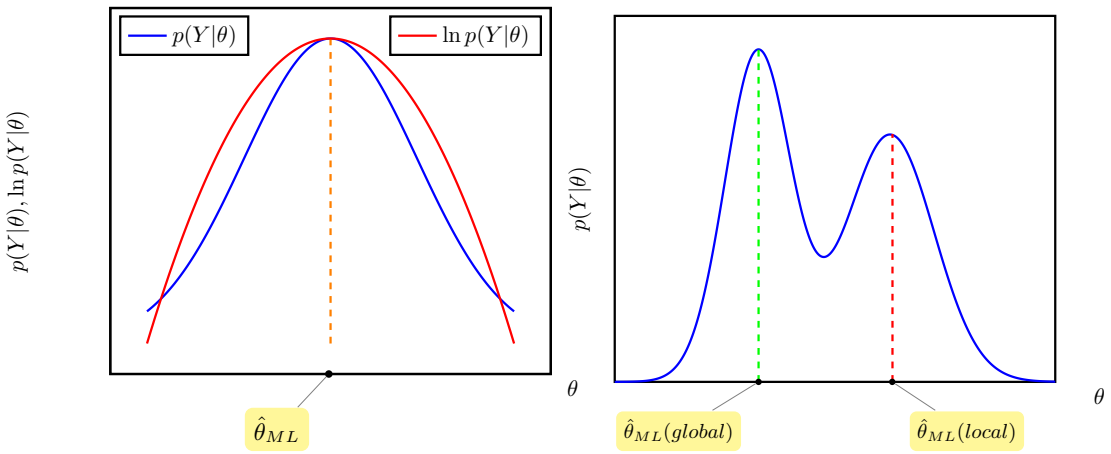
$$\hat{\theta}_{ML} = \arg\max_{\theta} \; p(Y|\theta). \tag{2.3}$$

In many cases, the likelihood function can be expressed analytically using standard functions, from which ML estimates can be easily computed by setting the first derivative to zero and solving for the parameters.

Log-likelihood function

When the likelihood function is described by an exponential function, using the log-likelihood function simplifies the ML estimation to a maximization of a polynomial function. As log is a monotonically increasing function, the ML estimate does not change. Therefore, (2.3) is equivalent to,

$$\hat{\theta}_{ML} = \arg\max_{\theta} \; \ln \; p(Y|\theta). \tag{2.4}$$



(a) *MLE in log-likelihood function* (b) *MLE in mixture likelihood*

Figure 2.3: *Illustration showing that ML estimation does not change when obtained from the likelihood and the log-likelihood functions. In the second figure, global and local ML estimates from a Gaussian mixture likelihood are depicted.*

Gaussian likelihoods are encountered commonly in many applications and obtaining ML estimate from such functions is straightforward, as the log of a Gaussian is a second order polynomial with a unique maximum [9]. The fact that ML estimate is the same for a Gaussian likelihood function and it's corresponding log-likelihood function is shown in Figure 2.3a for a scalar parameter case.

Mixture likelihoods

In multi-target tracking scenarios, multi-modal likelihood functions are commonly encountered. These are non-concave functions with multiple local optima. In ML context, this implies that more than one parameter setting can locally maximize the multi-modal likelihood function. This complicates the task of obtaining a global maximum point.

Mixture likelihoods can be conveniently handled by modelling them using Gaussian mixture models. But, finding ML estimates from these models poses a serious problem [7]. There are several strategies to obtain a unique (not necessarily global) ML estimate from mixture likelihoods; Expectation maximization is one widely used algorithm to approximate ML estimates, which will be discussed in detail in Chapter 6.

Figure 2.3b shows a univariate, multi-modal Gaussian mixture likelihood. By visual inspection, for this scalar parameter estimation case, it is easy to figure out the global ML estimate, which is not the case for high dimensional likelihood functions.

2.2.2 Bayesian parameter estimation

ML estimation is useful in many applications, but it does not have the scope to incorporate any prior information we might have about the parameters, as its performs the estimation only based on the observed data. While this might still yield good estimates, the estimates are entirely dependent on the quality of the observed data. In target tracking applications, some useful prior information about the whereabouts of the targets being tracked can be known before observing any data. Information like the scan region, approximate starting points and maximum velocity of targets can be known with certain degree of certainty. In such scenarios, a Bayesian approach

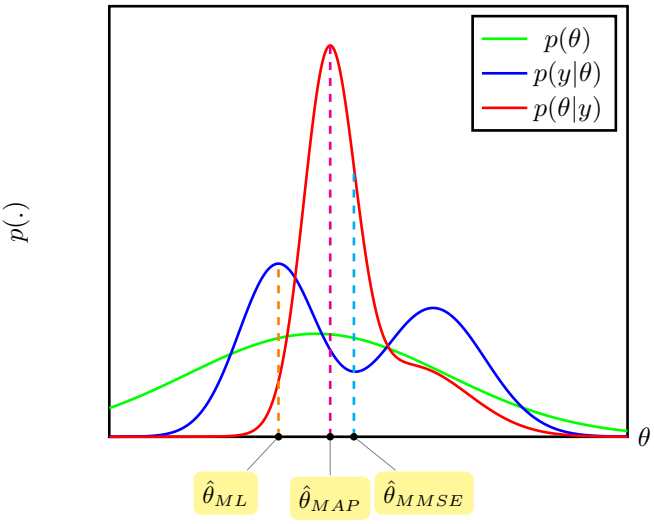


Figure 2.4: *ML, MAP and MMSE estimates.*

can benefit the solution to estimating parameters. Further, Bayesian estimation is less sensitive to the statistical noise in data [10].

In Bayesian parameter estimation, given a set of observations Y, we are still interested in estimating the unknown parameter θ, but in contrast to ML estimation, we model the θ as a random variable. As the parameters are treated as random variables, we can place a probability density to reflect our prior beliefs about the parameters, called the *prior* density. We then use information from the likelihood of observed measurements to obtain the updated density, commonly referred to as the *posterior* density. This idea is encapsulated in the simple, yet powerful Bayes' rule, as,

$$posterior \propto prior \times likelihood \tag{2.5}$$

$$p(\theta|Y) \propto p(\theta) \times p(Y|\theta). \tag{2.6}$$

The posterior density captures our updated beliefs using the information obtained from measurements. Bayesian estimators use thus obtained posterior density to obtain the parameters using two common estimators, which are discussed next.

Maximum a posteriori estimator

Maximum a posteriori (MAP) estimator, as the name suggests, chooses the parameter $\hat{\theta}$ that maximizes the posterior PDF,i.e, *mode* of the posterior PDF, written as,

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta|Y) \tag{2.7}$$

$$= \arg\max_{\theta} p(Y|\theta)p(\theta). \tag{2.8}$$

We see that, MAP estimator is similar to the ML estimator, but for the prior density on the parameters. Further, it can be shown that the MAP estimators minimize the "hit-or-miss" cost function [11].

Minimum mean squared error estimator

Choosing the mode of a heavily asymmetrical posterior distribution (for example, the posterior PDF in Figure 2.4) yields the best estimate only in MAP sense. In such cases primarily, the parameter that minimizes the mean squared error (MSE) might be better. The estimator that minimizes the MSE is called the *minimum MSE* (MMSE) estimator. MMSE minimizes the MSE cost function, and it can be shown that, by doing so, it yeilds the *mean* of the posterior PDF [11]. Thus,

$$\hat{\theta}_{MMSE} = \int \theta p(\theta|Y) d\theta \tag{2.9}$$

$$= \mathbb{E}(\theta|Y), \tag{2.10}$$

where the expectation is with respect to the posterior PDF.

In Figure 2.4, we depict the parameters chosen by the ML, MAP and MMSE estimators. The ML estimator, chooses the parameter that maximizes the likelihood function, whereas, the Bayesian estimators act on the posterior PDF; the MAP estimator chooses the highest mode of the posterior density and the MMSE estimator chooses the mean of posterior density.

Chapter 3

Bayesian filtering and smoothing

In the previous chapter, we discussed parameter estimation from a fixed batch of observed data. In many applications, it is often of interest to estimate parameters from measurements observed at regular intervals. This form of *sequential* estimation is a common occurence in dynamical systems, where the system state evolves with time. Estimation of target states in tracking applications is a typical scenario where the states evolve over time. In this chapter, we extend the notion of parameter Bayesian parameter estimation to sequential estimation problems. We discuss filtering and smoothing solutions, which can be used to recursively obtain parameters from batches of sequentially observed data. We present the recursive filtering and smoothing solutions, in the form of Kalman filter and RTS smoother, respectively, for linear and Gaussian model assumptions.

3.1 Sequential parameter estimation

If $\mathbf{x}_t$ are the parameters of interest that evolve over time, and $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$ correspond to all the measurements up to time t, then we are interested in a posterior distribution $p(\mathbf{x}|\mathbf{y})$, from which the MAP or MMSE estimate of parameters can be obtained at each time instant. We are interested in the marginal posterior density i.e, the density over parameters at one time instant, instead of the joint density over the parameters at all time instants. This in fact reduces the complexity of the estimation problem immensely [12]. Depending on the marginal posterior density we are interested in, sequential estimation becomes the following tasks:

- **Prediction**: As the name suggests, we are interested in the marginal posterior density of the future states conditioned on measurements up to the present time instant. The predicted posterior density is given by,

$$p(\mathbf{x}_{t+n}|\mathbf{y}_{1:t}), \quad n = 1, 2, \ldots \quad t = 1, \ldots, T. \tag{3.1}$$

- **Filtering**: It forms the most common type of sequential parameter estimation procedure, wherein, we are interested in obtaining a posterior density over parameters for a particular time instant, given all the measurements up to that time instant. The filtering posterior density is given as,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}), \quad t = 1, \ldots, T. \tag{3.2}$$

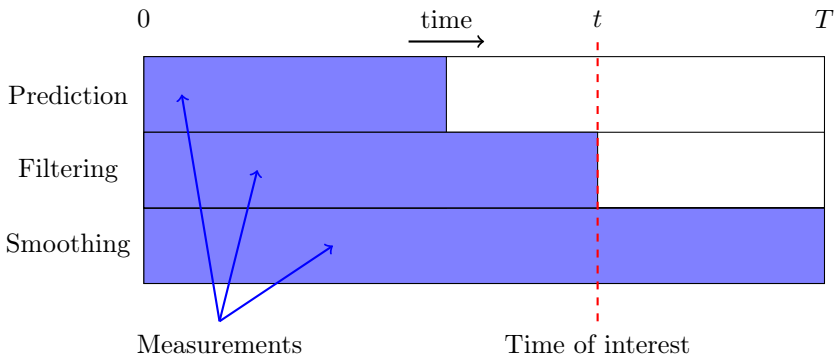


Figure 3.1: *Prediction, filtering and smoothing.*

- **Smoothing**: In smoothing, we are interested in a density over the parameters at a particular time instant, given measurements up to and beyond the time of interest. Intuitively speaking, this scenario conveys more information (future measurements) yielding improved estimates. The smoothed posterior density for the *fixed interval smoothing* is given as,

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}), \quad t = 1, \ldots, T. \tag{3.3}$$

In this thesis, we are mainly concerned with filtering and smoothing, and discourse about these follow in the remainder of this chapter. Figure 3.1 depicts the idea of prediction, filtering and smoothing in relation to the estimate we are interested in and the available measurements [12].

3.2 Bayesian filtering

In this section we focus on the most widely used Bayesian estimation strategy to evaluate parameters from sequential data - Bayesian filtering. The general problem of sequential data estimation is formulated by discussing the state space models and deriving the conceptual filtering solution. Closed form expression for Kalman filter, which is the most widely used Bayesian filtering algorithm for linear and Gaussian model assumptions is presented.

3.2.1 Problem formulation

Dynamical systems evolve over time with an inherent dependence between states acroos time. One widely used system model that can capture the state evolution of dynamical systems, like moving targets, is the first-order Markov model [13]. If $\mathbf{x}_t$ is the target state at time t, the first-order Markov assumption implies, that the current state of system $\mathbf{x}_t$ is independent of all other previous states, conditioned on the immediately previous state $\mathbf{x}_{t-1}$. This can be expressed as the conditional density,

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \ldots, \mathbf{x}_1, \mathbf{x}_0) = p(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{3.4}$$

The density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is usually called the state transition density, as it captures the transition between consecutive states. The transition between states is generally specified using the motion model, given as,

$$\mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}). \tag{3.5}$$

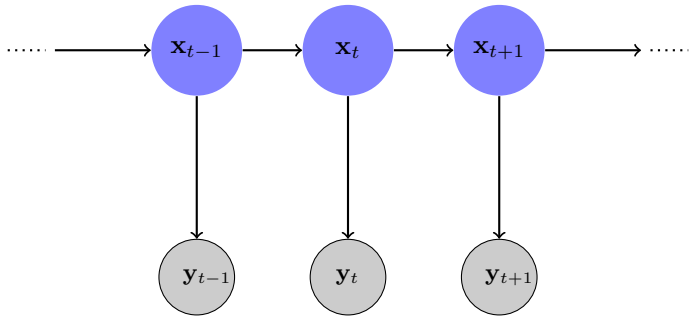


Figure 3.2: *Target state evolution with first-order Markov chain assumption.*

where $\mathbf{v}_{t-1}$ is the process noise.

Another commonly made assumption that simplifies dynamical systems is to assume that measurement at a particular time is dependent only on the target state at that time instant. This is a reasonable assumption to make, as the measurement from a target that is evolving according to (3.4), at any given time instant depends only on the current target state. This assumption is captured in the conditional probablility,

$$p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_1, \mathbf{x}_0) = p(\mathbf{y}_t|\mathbf{x}_t). \tag{3.6}$$

The term $p(\mathbf{y}_t|\mathbf{x}_t)$ is interpreted as the likelihood of the measurement $\mathbf{y}_t$, given that it was generated from the true state $\mathbf{x}_t$. Target states and measurements are related via the measurement model, given as,

$$\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t). \tag{3.7}$$

where $\mathbf{w}_t$ is the measurement noise.

These assumptions are ubiquitously used and found adequately expressive when representing many dynamical systems. The Bayesian network in Figure 3.2 depicts these assumptions, where the states are shown as circles to identify them as random variables and measurements using rectangles to indicate that they are observed.

In (3.2) we presented the posterior density, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, as the density of interest in filtering applications, using which MAP or MMSE estimates can be obtained as in (2.8) and (2.10). With the assumptions detailed in this section, we are ready to present the filtering equations in the next section.

3.2.2 Conceptual solution

Next, we show that the filtering posterior density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ can be computed recursively using the model assumptions in (3.4) and (3.6) and all measurements, $\mathbf{y}_{1:t}$.

The filtering density of interest at time t, from (3.2), can be rewritten as,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{y}_t, \mathbf{y}_{1:t-1}), \tag{3.8}$$

where we have split the conditioning over measurements into a set of measurements up to $(t-1)$ and only the measurements at t. Using Bayes' rule in the right hand side of the above equation, we get,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1})p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}. \tag{3.9}$$

The first factor in the numerator, $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1})$, can be written as the likelihood function, $p(\mathbf{y}_t|\mathbf{x}_t)$, based on the assumptions made in (3.6). The second factor in the numerator can

be identified as the prediction density, wherein, we are interested in the posterior at t, given measurements up to $(t-1)$, as discussed in (3.1). As we are interested in the posterior density over the variable $\mathbf{x}_t$, the denominator is only a normalization constant. Absorbing the denominator into the proportionality, (3.9) can be written as,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1}). \tag{3.10}$$

This is consistent with the Bayesian notion, where the posterior density on the left hand side is proportional to the product of the likelihood function and the prior density, which in this case is the predicted distribution.

At this point, only the predicted distribution is unknown, which can be written using the posterior density at $(t-1)$ and the transition density, which are assumed to be known. Using the following marginalization step,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \tag{3.11}$$

$$= \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \tag{3.12}$$

$$= \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}, \tag{3.13}$$

where, in the last step we have used the Markov property of state transition density, to remove the conditioning on $\mathbf{y}_{1:t-1}$. The expression in (3.13) is also known as the Chapman-Kolomogorv equation. Thus, we now have all components to perform filtering, which can be carried out in two steps:

- Prediction step: Compute the predicted density at time t using the posterior at $(t-1)$ and the transition density, using (3.13).
- Update step: Obtain the required filtering posterior density, by updating the predicted density with the likelihood of the measurement using (3.10).

In the next section, we present the Kalman filter, which is the most popular filtering algorithm obtained under linear Gaussian assumptions for the motion and measurement models.

3.2.3 Kalman filter

The conceptual solution presented in the previous section turns out to be intractable, unless the motion and measurement models are constrained to have certain amenable properties. To obtain a tractable filtering algorithm that can still be applied to a wide variety of problems, linear and Gaussian model assumptions are enforced on the motion and measurement models.

If the state vector $\mathbf{x}_t$ and measurement vector $\mathbf{y}_t$ comprise of m and n scalar states, respectively, using linear motion and measurement models, we get,

$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{v}_{t-1} \tag{3.14}$$

$$\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{w}_t, \tag{3.15}$$

where, $\mathbf{F_t} \in \mathbb{R}^{m\times m}$ and $\mathbf{H_t} \in \mathbb{R}^{n\times m}$. The motion noise and measurement noise are modeled as zero-mean Gaussian random variables, $\mathbf{v}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, with covariances $\mathbf{Q}$ and $\mathbf{R}$, respectively. Further, if the prior density at time zero, $p(\mathbf{x}_0)$ is assumed to be Gaussian,

it can be shown that the posterior density obtained at subsequent time instants will also be a Gaussian density, i.e, the posterior density at t will be of the form,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}), \tag{3.16}$$

where $\hat{\mathbf{x}}_t$ is the mean and $\mathbf{P}_{t|t}$ is the covariance of the posterior density at t. As any Gaussian density is completely described by its mean and covariance, the filtering posterior density in Gaussian form is tractable under the linear and Gaussian assumptions.

The Kalman filter [14] provides closed form expressions for the mean and covariance of the various involved densities in the prediction and update steps, which are summarized below.

Kalman filter prediction

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t-1}\hat{\mathbf{x}}_{t-1|t-1} \tag{3.17}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_{t-1}\mathbf{P}_{t-1|t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1}. \tag{3.18}$$

Kalman filter update

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{y}}_t \tag{3.19}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T. \tag{3.20}$$

where, $\tilde{\mathbf{y}}_t$ is the new information provided by the measurement, commonly called the innovation. The term $\mathbf{K}_t$ is the Kalman gain determines the importance that should be given to the innovation, and $\mathbf{S}_t$ can be interpreted as the predicted covariance of the measurement $\mathbf{y}_t$. Expressions for each of these terms are given by,

$$\tilde{\mathbf{y}}_t = \mathbf{y}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1} \tag{3.21}$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1} \tag{3.22}$$

$$\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t \tag{3.23}$$

The Bayesian intuition that prediction should increase uncertainty, and incorporation of new information must reduce one's uncertainty is clearly captured in the expressions for $\mathbf{P}_{t|t-1}$ and $\mathbf{P}_{t|t}$ above. Kalman filter is both – the MAP and MMSE estimator, as the mode and mean of the (Gaussian) posterior density are the same. This makes Kalman filter further appealing.

Kalman filter is the workhorse filtering algorithm (for linear and Gaussian assumptions), and it has been used extensively to perform all the filtering discussed in this thesis. Next, we briefly look at optimal smoothing, and using similar assumptions used to arrive at Kalman filter, we present a tractable smoothing algorithm.

3.3 Optimal smoothing

In this section, we discuss smoothing, wherein, we are interested in the problem of estimating parameters at each time, conditioned on all the measurement available. This is in contrast to filtering, where we were interested in estimating parameters at current time, given measurements up to that time. We present the exact smoothing equations only for the fixed-interval smoothing, and summarize the closed form expressions for a smoother under linear and Gaussian model assumptions.

There are three main versions of smoothing, depending on the amount of observed data that is available and the point of interest where the parameters are to be estimated.

1. **Fixed interval smoothing**: The density of interest is $p(\mathbf{x}_t|\mathbf{y}_{1:T})\,\forall\, t$, and a fixed T.
2. **Fixed lag smoothing**: The density of interest is $p(\mathbf{x}_t|\mathbf{y}_{1:t+n})\,\forall\, t$, and a fixed $n > 0$.
3. **Fixed point smoothing**: The density of interest is $p(\mathbf{x}_t|\mathbf{y}_{1:T})$, at a fixed t, as T grows.

We focus the discussion mainly on the fixed interval smoothing, implemented using *forward-backward smoothing* (FBS). This form of smoothing is most suitable, when dealing with target trajectories of fixed length.

3.3.1 Conceptual solution

The density of interest in fixed interval smoothing is $p(\mathbf{x}_t|\mathbf{y}_{1:T})\,\forall\, t = 1,\ldots,T$, where T is the fixed interval. In this section, we present the exact solution to this smoothing problem using the FBS approach.

Forward-backward smoothing, as the name suggests, involves performing a forward operation (filtering) and a backward operation (smoothing). The aim of FBS is to:

- Compute $p(\mathbf{x}_t|\mathbf{y}_{1:t})\,\forall\, t = 1,\ldots,T$, according to the optimal Bayesian filtering equations in (3.13) and (3.10).
- Compute $p(\mathbf{x}_t|\mathbf{y}_{1:T})\,\forall\, t = T-1,\ldots,1$, using the optimal backward smoothing equation [15],

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}) = p(\mathbf{x}_t|\mathbf{y}_{1:t}) \int \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})}{p(\mathbf{x}_{\mathbf{t+1}}|\mathbf{y}_{\mathbf{1:t}})} d\mathbf{x}_{t+1}. \tag{3.24}$$

where, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is the filtering distribution at time t.

3.3.2 Rauch-Tung-Striebel smoother

As with Bayesian filtering, closed form expressions for optimal smoothing can be obtained for linear and Gaussian model assumptions. In case of the FBS, these assumptions result in the *Rauch-Tung-Striebel*(RTS) smoother, which employs Kalman filtering to perform the forward filtering and has similar closed form expressions for backward smoothing.

With the model assumptions in (3.5) and (3.7), the smoothed posterior distribution is Gaussian, $p(\mathbf{x}_t|\mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|T}, \mathbf{P}_{t|T})$ [12]. The RTS smoother equations for the forward filtering (same as Kalman filtering equations) and backward smoothing operations, to obtain $\hat{\mathbf{x}}_{t|T}$ and $\mathbf{P}_{t|T}$ for all $t = T-1,\ldots,1$, are given below.

Forward filtering

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1} \tag{3.25}$$
$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}) \tag{3.26}$$
$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T. \tag{3.27}$$

Backward smoothing

$$\mathbf{G}_t = \mathbf{P}_{t|t}\mathbf{F}_t^T\mathbf{P}_{t+1|t}^{-1} \tag{3.28}$$
$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{G}_t(\hat{\mathbf{x}}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{3.29}$$
$$\mathbf{P}_{t|T} = \mathbf{P}_{t|t} - \mathbf{G}_t(\mathbf{P}_{t+1|t} - \mathbf{P}_{t+1|T})\mathbf{G}_t^T. \tag{3.30}$$

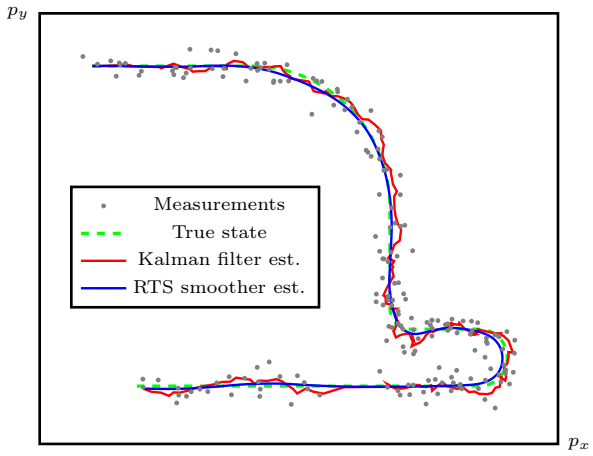


Figure 3.3: *Single target state estimation using Kalman filter and RTS smoother.*

Bayesian smoothing provides improved estimates than with only Bayesian filtering, as any smoother typically utilizes more information than a filter. This is illustrated using an example of trajectory estimation of an object, maneuvering in a difficult path, shown in Figure 3.3. The Kalman filter estimates are jagged, as the filter estimates the target state using measurements only up to that time. The RTS smoother estimates are evidently less noisy, or smoothed. This smoothness can be attributed to the use of additional information; to estimate parameters at any instant, RTS smoother uses all the information from the entire interval of measurements. In applications where some lag is bearable, smoothing can yield improved estimates than an online filter.

Chapter 4

Bayesian multi-target tracking

With the background on Bayesian filtering and smoothing established in the previous chapter, we now present the basics of *multi-target tracking* (MTT). While Bayesian filtering is an essential part of the solution when dealing with MTT, the core of the problem that will be discussed in detail here pertains to the challenges involved in data association. The uncertainties that arise in associating measurements to targets is the problem mainly addressed in this thesis. A generic MTT system with blocks corresponding to different tasks is first presented. It is followed by description of two popular data association strategies – *global nearest neighbour* (GNN) based on hard-decisions, and the soft-decisions based *joint probabilistic data association* (JPDA).

4.1 Multi-target tracking

When Bayesian filtering is used in target tracking applications, it is implicit that a single object is being tracked *without* any uncertainty in the measurement origin. It is assumed that a set of measurements are provided for the entire length of the trajectory (or for the most part of it). As mentioned in Chapter 3, the Kalman filter provides an analytical expression for the posterior distribution. Under the appropriate model assumptions (linear and Gaussian motion and measurement models) Kalman filter can be used; filters like the extended Kalman filter [12] are useful when dealing with non-linear models. Filtering is a simpler problem than target tracking, in the sense that it is what remains once we have resolved the data association and track maintenance problems.

- **Data association uncertainties**: Working out measurement-to-target associations is an important first step in target tracking. After the data association problem is solved, measurements can be partitioned to belong to individual targets and then used for trajectory estimation by performing filtering.
- **Track maintenance**: In many applications, the number of targets being tracked is not fixed, and is unknown. In such scenarios, the tracking solution must be able to handle appearance and disappearance of targets as tracking is in progress. This amounts to the problem which is called track maintenance.

The task of track maintenance is briefly addressed in Section 4.1.1, while the challenges posed by data association uncertainties are discussed to a greater detail in Section 4.2.

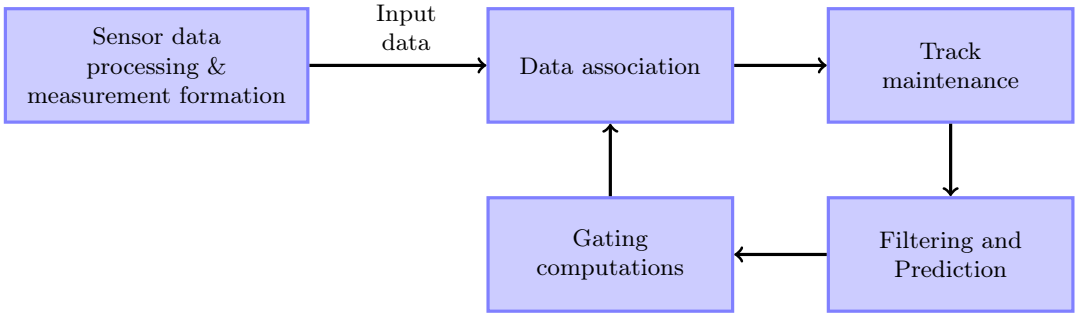


Figure 4.1: *A generic multi-target tracking system.*

An overview of a typical MTT system is shown in Figure 4.1 from [16]. Except the sensor data processing block, the remaining aspects of MTT are discussed in this thesis. The primary contributions focus on solving data association related challenges.

To gain intuition of the specific problem at hand and to acquaint with the existing solution methods, a typical scan region along with the true tracks of targets in an MTT scenario as seen by a sensor are shown in Figure 4.2. As can be seen in the figure, it is extremely difficult to visually inspect and spot the trajectories present in the scan region, which also translates into complications when solved using tracking algorithms. In the remainder of the chapter, tractable MTT solutions that solve many such interesting multi-target tracking problems are discussed. The optimal solution is presented, along with strategies like gating, pruning and merging of hypotheses that substantially reduce complexity.

4.1.1 Track maintenance

In this section, one of the distinguishing features between filtering and tracking – track maintenance is presented. In multi-target tracking applications, it is common to encounter scenarios where targets appear and disappear in the scan region. Appearance and disappearance, also referred to as birth and death, of targets are handled by the track maintenance block of the MTT algorithm. In traditional algorithms, track maintenance includes the following steps:

- **Initiation**: At each scan, the multi-target tracking algorithm must look for possible birth of new targets. This is accomplished in many ways – one common way is to pretend all received measurements are new, to create hypotheses with new tracks initiated in them and then rely on the track confirmation logic to make a decision.

- **Confirmation**: After initiating tracks from measurements, the newly formed tracks can be observed for a certain number of scans; by the end of which (in most cases) possible confirmed tracks can be obtained based on the track score (likelihood-based measure). Thus, only the confirmed tracks are maintained, while the remainder of them are deleted.

- **Termination**: Apart from terminating candidate tracks that are confirmed to be not from targets, whenever targets stop existing, they must be marked as terminated and must not be featured in the subsequent tracking steps. One ad-hoc way of terminating tracks is to check if the target is not generating measurements, i.e, if it has gone undetected for consecutive scans. If there are no measurements being associated to a target for a certain number of predetermined scans, it can be declared as terminated.

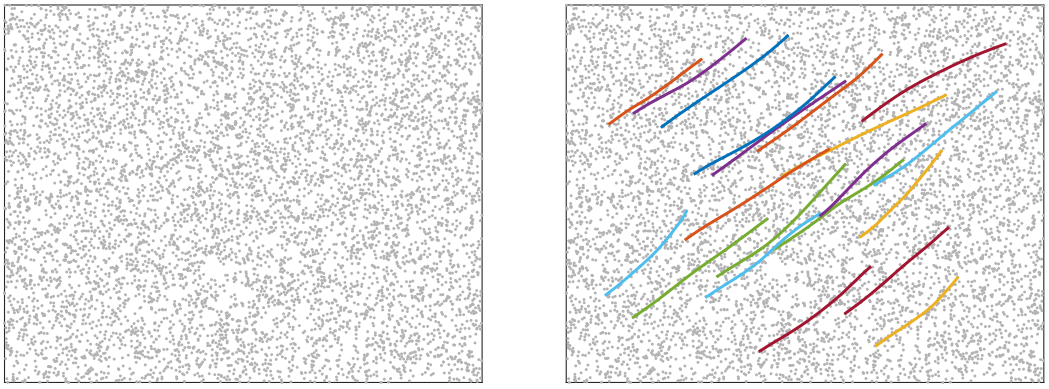


Figure 4.2: *(**left**) A scan volume with target measurements from all time scans in clutter. (**right**)True trajectories of targets to be tracked.*

In this thesis, the focus is primarily on the data association aspects of the problem and track maintenance is not handled. Thus, it is assumed there are known and fixed number of targets, at all time scans.

4.2 Data association in target tracking

In this section, the challenges faced in resolving measurement-to-target association uncertainties are discussed. This problem is encountered in various other domains as well, and is commonly termed as the problem of data association. Due to the inherent need for simultaneous tracking of many targets in difficult environments, the problem of data association is all the more pronounced in multi-target tracking applications. Various causes leading to data association uncertainties are elaborated here. After establishing the challenges, gating is discussed as a possible technique that reduces data association complexity. Commonly used data association strategies, which also find wide applicability in the current thesis work – the single hypothesis methods like the GNN and JPDA filter are discussed.

4.2.1 Uncertainties in measurements

Data association uncertainties are caused due to multiple reasons; noisy measurements and clutter measurements being the most common and important of them. In point target tracking, it is further aggravated as the measurements are points. Consequently, the point measurements usually do not carry any descriptive features specific to the targets they were generated from. This is in contrast with sensors that can track extended targets, where some additional features can be extracted from the sensor measurements. For instance, when using cameras as sensors, features like edges, contours and even behaviour of the targets can be extracted, which can be useful when attempting data association [17]. Three important contributing factors to measurement uncertainties are presented next.

Noisy measurements

Measurements obtained from sensors are not perfect. They are corrupted by noise arising from the noise in electrical signals and/or due to ambient noise. As described in Chapter 3, Bayesian filtering deals with measurement noise by modelling it as a random variable, according to the measurement model (3.7):

$$\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t),$$

where, $\mathbf{w}_t$ is the measurement noise. In many applications, it is common to model the measurement noise as being additive, and Gaussian with zero mean and covariance, $\mathbf{R}$.

Clutter detections

Sensors pick up measurements indiscriminately from the scan volume from targets and non-targets, as there is no intelligence at the sensor level to perform this distinction. This leads to the problem of observing spurious measurements from non-target objects, which are referred to as clutter measurements. Clutter measurements are unavoidable in real-tracking applications and must be dealt with.

There are implementations of MTT algorithms, where every measurement must be assigned to a target to satisfy the model assumptions being used. In such cases, clutter measurements that fall within the gate of a predicted target measurement are assigned to a dummy track, and the related hypotheses are discarded in subsequent scans. It is common in tracking applications to assume that the clutter measurements are uniformly distributed in the scan region of volume, V, and the number of clutter measurements,M_c, is assumed to be Poisson distributed with intensity β_c. Therefore, the cardinality of clutter measurements is distributed as,

$$P_c(M_c) = \frac{\exp(-\beta_c V)(\beta_c V)^{M_c}}{M_c!}. \tag{4.1}$$

Undetected targets

Depending on the sensor resolution and quality of environment (dense clutter, high noise), there are chances of the sensor missing target-generated measurements. When targets are in close proximity, a sensor with poor resolution might not be able to resolve all the targets. Another reason for missing target detections could be when the measurement noise is higher than the signal noise – this can happen due to attenuation of the signal noise, or increase in the measurement noise. Also, targets can be missed when they enter shadow region of a sensor's field of view. These uncertainties in detecting target generated measurements are modelled as the probability of detection, P_D. When $P_D = 1$, it implies that all target-generated measurements are received at each scan. On the other hand, lower value of P_D implies some of the targets could go undetected at each time scan.

4.2.2 Conceptual solution

In this section, the conceptual data association solution for the case of point targets is presented under commonly used assumptions. The model assumptions are presented next.

Model assumptions

- Each target generates at most one measurement.
- Each measurement can be assigned to at most one target.

- All targets evolve according to the same motion model, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$.
- Targets evolve independently of each other.
- All measurements are generated according to the same measurement model, $\mathbf{p(y_t|x_t)}$.
- Number of targets is known, denoted as N.

The first two assumptions make the problem specific to point target tracking, by ensuring that no target is assigned to multiple measurements, and that no measurement is assigned to multiple targets. This is a reasonable constraint, as every target generates only one measurement in point target scenarios. The remaining assumptions simplify the problem to some extent.

Following the notation that is common in target tracking, $\mathbf{X}$ denotes target states of all targets at all time instants. It is represented as a multi-dimensional vector, $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_N]$. Elements of $\mathbf{X}$ are individual target states, $\mathbf{X}_i = [\mathbf{x}_{1i}, \ldots, \mathbf{x}_{Ti}]$, where, $\mathbf{x}_{t,i}$ is state of target i at time t, and T is total number of scans. Likewise, measurements are denoted as vectors, $\mathbf{Y} = [\mathbf{Y}_1, \ldots . \mathbf{Y}_T]$, where each $\mathbf{Y}_t$ corresponds to the M_t measurements obtained at time t. Following these notations, the objective of multi-target tracking is to obtain the MAP or MMSE estimates of target states from the posterior density $p(\mathbf{X}|\mathbf{Y})$.

When the measurements and data association information are given, the posterior density can be obtained by performing Bayesian filtering or smoothing. Conversely, the posterior density, $p(\mathbf{X}|\mathbf{Y})$, can be tractably expressed only when the data association is known. One of the common approaches to resolving data association uncertainties is to model data association uncertainties as a discrete random variable, and obtaining a posterior density conditioned on the best data association hypothesis. There also exists a class of algorithms, where the problem of data association is handled implicitly by using a random-finite set approach [18] [19], which is beyond the scope of this thesis. The focus here is on solutions which use explicit data association variables to resolve uncertainties.

A discrete random variable, $\mathbf{K}$, is introduced to represent the data association uncertainties, such that each instance of $\mathbf{K}$ corresponds to a global hypothesis. A global data association hypothesis is a mapping of target indices to measurement indices over time, such that it satisfies the aforementioned point target data association assumptions. If $\mathbf{k}$ is a global hypothesis instantiated from $\mathbf{K}$, it can be represented in matrix form, such that its rows are vectors for each target i, given as, $\mathbf{k}_i = [k_{1,i}, \ldots, k_{T,i}]$. The elements $k_{t,i}$ contain the measurement index assigned to the target i, at time t, i.e,

$$k_{t,i} = \begin{cases} j & \text{if measurement } j \in \mathbf{Y}_t \text{ is associated to target } i, \\ 0 & \text{if target } i \text{ is undetected.} \end{cases} \tag{4.2}$$

Conditioned on an instance of $\mathbf{K}$, the required posterior density becomes, $p(\mathbf{X}|\mathbf{Y}, \mathbf{K} = \mathbf{k})$. This density can be estimated using Bayesian filtering/smoothing techniques. The original posterior density can be calculated by summing over all possible data association hypotheses, as

$$p(\mathbf{X}|\mathbf{Y}) = \sum_{\mathbf{K}} \Pr\{\mathbf{K} = \mathbf{k}|\mathbf{Y}\} p(\mathbf{X}|\mathbf{K} = \mathbf{k}, \mathbf{Y}). \tag{4.3}$$

The above equation can be interpreted as the weighted sum of posterior densities, conditioned on different global hypotheses. The weight in the above expression is the probability of data association hypothesis, $\Pr\{\mathbf{K} = \mathbf{k}|\mathbf{Y}\}$. As we have seen in Chapter 1, the number of hypotheses grows exponentially with the number of measurements over time, making the computation of (4.3) intractable. All multi-target tracking algorithms attempt to approximate the true posterior to yield a tractable solution.

One useful observation about the form of the posterior density in (4.3) is, it is a *mixture model* with possibly several local optima. For the linear and Gaussian assumptions for motion and measurement models, the true posterior density is the sum of weighted Gaussian densities i.e, a Gaussian mixture model (GMM) discussed in Section 2.1.2. As a result, the task of obtaining MAP estimate of target states from the true posterior in MTT applications culminates into a problem of solving for the global maximum in a multi-modal Gaussian mixture model. This problem is inherently complicated for several reasons: an important one being the presence of several local optimal points. The challenge of dealing with local optima was touched upon when discussing ML estimates for mixture likelihoods in Section 2.2.1.

4.2.3 Gating

Gating is a hard-decision technique used to reduce the number of measurement-to-track associations [16] in densely cluttered tracking applications. Measurements with negligible probability of being generated by an existing target are preemptively discarded by performing gating; this avoids the additional computation due to the hypotheses involving these unlikely measurements. Consider the case of radar-based tracking of an aircraft, moving in a straight line. It is a reasonable assumption that the aircraft could not have generated measurements defying laws of physics. For instance, there is very little chance of obtaining measurements from the scan region far behind the aircraft, or that it can be spotted in a region beyond what it could have traversed within subsequent scan intervals. Gating excludes such unlikely measurements based on the motion model of the targets being tracked. Thus, reducing the complexity to a considerable extent. There exist two common gating strategies – rectangular gating and ellipsoidal gating, differing mainly in their complexity, apart from the obvious difference in their shape.

Rectangular gating

One basic gating strategy is to discard measurements that are outside a broad rectangular region, centered around the predicted measurement. This form of rectangular gating is crude, but finds use in densely cluttered scenarios due to its cheap computational cost. Denoting $\mathbf{y}_{t|t-1}$ to be the predicted measurement with $x-y$ components, $[x_{t|t-1}, y_{t|t-1}]^T$, and $\mathbf{y}_t$ as the received measurement with components, $[x_t, y_t]^T$. The decision if the measurement falls within the rectangular gate is based on the simple thresholding condition:

$$|y_t - y_{t|t-1}| \leq \kappa\sigma^y_{t|t-1}, \quad |x_t - x_{t|t-1}| \leq \kappa\sigma^x_{t|t-1}, \tag{4.4}$$

where, $\sigma^y_{t|t-1}$ and $\sigma^x_{t|t-1}$ are the covariances of the predicted measurement in x and y dimensions, discussed in (3.23). The rectangular gating coefficient is usually set, $\kappa > 3$, for desirable performance [16].

Ellipsoidal gating

Rectangular gating is usually followed by ellipsoidal gating, which utilizes the predicted measurement covariance $\mathbf{S}_t$, from (3.23), to create ellipsoids in the scan volume to further remove unlikely measurements. This form of gating is more computationally expensive (due to a matrix inversion operation), but yields finer set of measurements than the coarse set obtained from rectangular measurements. The ellipsoidal gating condition is given as:

$$(\mathbf{y}_{t|t-1} - \mathbf{y}_t)^T \mathbf{S}_t^{-1} (\mathbf{y}_{t|t-1} - \mathbf{y}_t) \leq \mathcal{G} \tag{4.5}$$

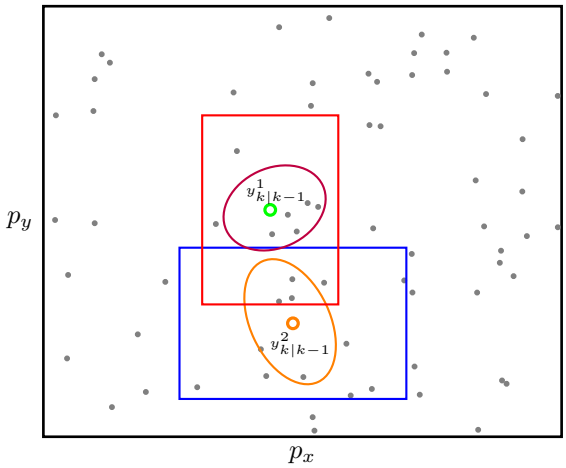


Figure 4.3: *Illustration of rectangular and ellipsoidal gating of measurements (gray points), for the case of two predicted measurements shown as circles.*

where, $\mathcal{G}$ is the ellipsoidal gating threshold. It can be shown that the statistic term on the left hand side follows the χ^2-distribution [20], and the probability that a 2-dimensional measurement would fall within the gating threshold $\mathcal{G}$ is given as [16],

$$P_G = 1 - \exp(-\mathcal{G}/2). \tag{4.6}$$

Figure 4.3 shows the operation of rectangular and ellipsoidal gating for two predicted measurements. The measurements that fall within the ellipsoidal gate are the ones used to form subsequent data association hypotheses, instead of investing computation on hypotheses which would turn out to be of little or no significance.

4.3 Global nearest neighbour

Global nearest neighbour [16] is one of the simplest data association algorithms based on pruning of hypotheses at each time instant. It is quite effective in containing the explosion in number of data association hypotheses. GNN retains a single global data association hypothesis, by assigning the best candidate measurement to each target, every time scan. The best assignment is chosen based on the likelihood, given as,

$$\ell_{ij} = \ln p(y_j|\hat{y}^i_t), \tag{4.7}$$

where, ℓ_{ij} is the log of the likelihood that measurement y_j was generated by the predicted measurement from target i. For each target, GNN compares likelihood of all the measurements within its gate and assigns the measurement with the highest likelihood value. When there are measurements overlapping between target gates, there is a chance of contention for the same measurements. These conflicts can be resolved by looking for globally best assignments, thus satisfying the point target assumptions of assigning unique measurement to each target and *vice versa*. This global constraint based assignment allows GNN to be formulated as a 2-D assignment problem, with the task of obtaining the best associations that maximize the overall cost (log-likelihood) of such an assignment [21]. Table 4.1, shows an illustrative scenario of assigning three measurements to three targets. To demonstrate the contention for measurements, the example assumes all three measurements appear in all target gates. The result as seen is

	Measurement 1	Measurement 2	Measurement 3
Target 1	-1	-10	-8
Target 2	-4	-12	-7
Target 3	-10	-5	-15

Table 4.1: GNN assignment for the case of three targets and three measurements is depicted. Entries in the table are the log-likelihood values. The optimal assignment that maximizes the overall assignment likelihood is marked in blue circles. Although, Measurement 1 is the best assignment for Target 2, due to the constraint that a measurement must be assigned at most to one target, Target 2 is assigned to Measurement 3.

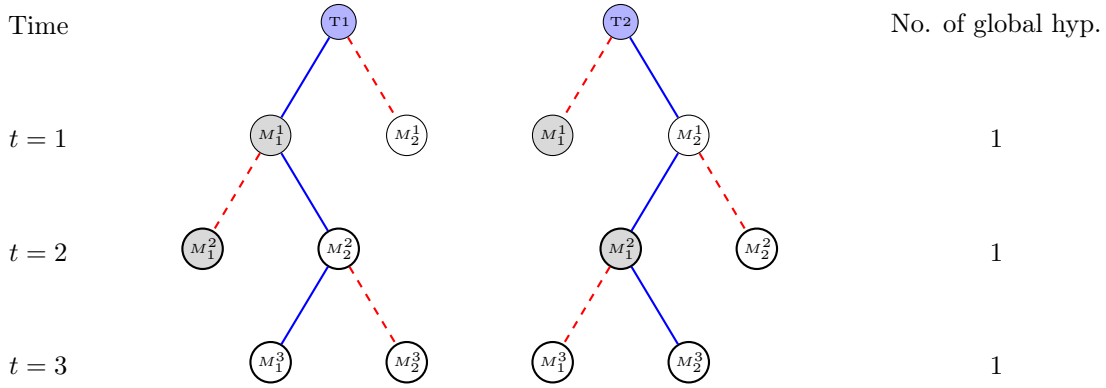


Figure 4.4: *Hypothesis tree for GNN. Blue solid lines are the two local hypothesis at each time scan, forming the single global data association hypothesis. The red dashed lines correspond to the hypotheses that have been pruned out.*

the best possible global assignment, as not all targets are assigned to their best local candidate measurement.

In terms of the hypothesis tree, GNN propagates a single global data association hypothesis, which is shown in Figure 4.4. The tree shows GNN operation over three scans for a simple two target tracking case. At $t = 1$, targets T_1 and T_2 are assigned to M_1^1 and M_2^1, respectively, and this is the only existing hypothesis when measurements at $t = 2$ arrive. Thus, GNN maintains only a single global hypothesis, in contrast to the optimal tree in Figure 1.1, where the number of hypotheses were growing exponentially for the same scenario.

GNN is simple, but makes hard decisions which degrade the quality of the solution. Nevertheless, it is one of the widely used data association solutions due to its simplicity. GNN is used as a starter for other algorithms to obtain initial set of data associations, which are subsequently improved by other means. Both of the proposed EM-based solutions in this thesis use GNN as the starter algorithm to obtain initial data associations, and optimize upon the thus obtained tracks. GNN is also an important component in the algorithm proposed in the Paper B, wherein it is run for all scans of data in several iterations.

4.4 Joint probabilistic data association

Making hard-decisions leads to information loss, as all but one hypothesis are discarded, when compared to making a soft (probabilistic) decision for the same scenario. This is the main motivation behind JPDA, which can be interpreted as the soft-decision counterpart of GNN [16]. JPDA considers all possible hypotheses using the measurements within the gate of a target and merges the hypotheses to obtain a single hypothesis instead of pruning as in GNN, which can be viewed as soft-decision. JPDA can be interpreted as an algorithm handling multiple hypotheses per time instant, but single hypothesis between time instants [22], [23]. Hypothesis merging is carried out using moment matching [24], which is discussed further in Chapter 6. This form of merging in JPDA is equivalent to updating the predicted target states with the *weighted* sum of measurements within the target's gate. These weights are the marginal probabilities of a particular measurement being assigned to a target, when considered across different global hypotheses.

Table 4.2 illustrates the computation of marginal data association probabilities for the case of three targets and three measurements. For this scenario, there are six possible global hypotheses, with their corresponding measurement-to-target associations shown in the table. For the purpose of this discussion, the posterior probability of each of the global hypotheses are assumed to be given (shown in the last column), based on computations in [22]. We are then interested in the marginal probability of a measurement j being assigned to target i, denoted as w_{ij}, which can be obtained by summing over all global hypotheses with this assignment. For instance, w_{12} in the table, corresponds to the marginal probability of assigning target 1 to measurement 2. This assignment occurs in two global hypotheses, H_3 and H_4. Thus, the marginal probability w_{12} is the sum of probabilities of these two hypotheses. In a similar way, marginal probabilities for all target and measurement assignments can be computed from the table. These marginal data association probabilities are used to compute the composite measurements used in updating target predicted states.

Hyp.	T1	T2	T3	Hyp. Prob.
H_1	1	2	3	0.35
H_2	1	3	2	0.1
H_3	2	1	3	0.25
H_4	2	3	1	0.05
H_5	3	1	2	0.15
H_6	3	2	1	0.2

$w_{11} = 0.35 + 0.1 = 0.45$

$w_{12} = 0.25 + 0.05 = 0.3$

$w_{13} = 0.15 + 0.2 = 0.35$

Table 4.2: Illustration of marginal probabilities calculation in JPDA

Computing marginal data association probabilities for this case seemed simple. In general, when dealing with large number of targets and measurements, this brute force approach of enumerating probabilities and summing over them is not very efficient. There are several techniques that can be used to approximate the marginal data association probabilities. *Loopy belief propagation* (LBP) in [25] is the preferred method adopted in this thesis, as the algorithm proposed in Paper A requires computing marginal data association probabilities. LBP uses the approach of passing messages over a graph with loops, to approximate marginal probabilities.

In some scenarios, JPDA is better than GNN [26], as it does not make hard decisions at each time instant. Nonetheless, as it approximates the posterior density by moment matching, it faces

problems in certain scenarios, leading to coalescence of tracks [27]. We investigate this behaviour in comparison to the proposed algorithm in Paper A, and show how the proposed algorithm is not faced with this problem.

4.5 Mixture model posterior density interpretation

In deriving the conceptual data association solution, it was shown that the true posterior density in (4.3) is multi-modal, which, for the linear and Gaussian model assumptions is a GMM. Using the GMM intuition, the various approximations made in GNN and JPDA to the true posterior are discussed next.

The single hypothesis strategy in GNN, based on picking the best global assignment amounts to approximating the mixture model posterior density at each time instant with its maximum mode. Thus, GNN approximates the true posterior at each time instant by its single largest component. In JPDA, the GMM is approximated with a single Gaussian using moment matching at each time instant. Both, GNN and JPDA, make approximations at each time instant based only on the measurements of that instant. There are other approaches studied in literature that can perform better by using measurements from multiple scans, when approximating the true posterior. *Multiple-hypothesis tracking* (MHT), which uses measurements from multiple scans to approximate the true posterior was one of the first MTT algorithms. MHT under some settings can provide optimal solution. In the next chapter, the track-oriented version of MHT, which serves as the benchmark algorithm in this thesis is discussed.

Chapter 5

Multiple hypothesis tracking

Single hypothesis methods like GNN and JPDA were discussed in the last chapter. Limitations in each of them, caused due to pruning of hypotheses in GNN and merging hypotheses in JPDA were highlighted. In this chapter, *multiple hypothesis tracking* (MHT) is discussed, which as the name suggests, maintains multiple hypotheses and picks the hypothesis with the maximum posterior probability. There are two main variants of MHT – the hypothesis oriented MHT and the *track–oriented MHT* (TOMHT), based on the nature of hypothesis tree that is maintained. The focus in this discussion will be primarily on TOMHT, in particular, the N–scan pruning version of it. The emphasis on TOMHT is for two reasons – the algorithm proposed in Paper B is philosophically similar to TOMHT, and it is used as the benchmark to compare the performance of the proposed EM–based solutions.

5.1 MHT – the optimal solution

MHT was first proposed in [28], and was one of the earliest algorithms to promote a tree–based approach in multi–target tracking that is still widely used. The primary difference between single hypothesis methods like GNN and JPDA is, MHT does *not* instantaneously decide the measurement–to–target assignments. Instead, MHT attempts to maintain multiple hypotheses and fixes the data associations based on measurements from multiple scans. Whenever this approach is tractable (for few targets and/or few scans), it can lead to the optimal data association solution discussed in Section 4.2.2. For reasons already discussed in Chapter 4, the optimal solution is not tractable; consequently, MHT employs strategies like pruning, but with an improvement called the *deferred decision* to obtain improved performance.

MHT defers the data association decision for the current time instant to a subsequent time instant, and maintains all possible hypotheses based on measurements obtained within this interval. At time instant $t+n$, for any scan depth $n > 0$, MHT considers all hypotheses based on n scans of measurements. It makes a hard decision on the data association at t that gives rise to the data association hypothesis with highest posterior probability at $t+n$. By doing this, the adverse effects of instantaneous data association can be considerably alleviated, as a data association decision after having seen subsequent measurements can be more reliable. Based on several existing MHT variations, this deferred decision approach has shown to yield improved performance, when compared to many other algorithms, even when tracking in densely cluttered environments [29].

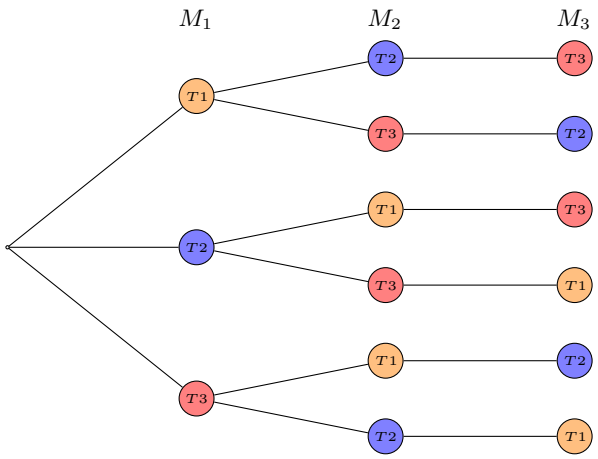


Figure 5.1: *Hypothesis tree for the hypothesis oriented MHT at one time instant, for the case of three targets and three measurements. Each branch of the tree is a valid global hypothesis.*

5.1.1 Hypothesis–oriented MHT

In hypothesis–oriented MHT, global data association hypotheses are groomed based on measurements received at each time instant. A global data hypothesis ensures that all tracks are compatible with each other. This was the original formulation of MHT in [28]. At each time instant, every existing hypothesis is expanded to incorporate all possible measurements and each of these newly formed hypotheses is then a valid global hypothesis. The number of hypothesis grows exponentially, and renders the solution intractable. The growth of hypotheses can be limited by performing some form of pruning. One popular method to curb the number of hypotheses is Murty's k–best algorithm [30], which ranks the k best hypotheses, which are the only ones propagated to next time instant. In Figure 5.1, a typical tree for the hypothesis–oriented MHT is shown for three targets and three measurements at one time instant. The first branch captures the hypothesis that assigns measurements M_1, M_2, M_3 to targets $T1, T2, T3$, respectively, which is a valid global hypothesis. As each branch in the tree is a valid global hypothesis, this variant of MHT is rightfully called the hypothesis oriented MHT.

5.1.2 Track–oriented MHT

Track–oriented MHT (TOMHT) also maintains hypothesis trees. But, as the name suggests, the tree describes individual target tracks rather than global hypotheses. For a single target case, the track tree is also the global hypothesis tree. In TOMHT, each branch of the hypothesis tree corresponds to a sequence of measurements across time describing a track. The tracks of individual targets that are compatible form a valid global hypothesis. The global hypothesis with the highest posterior probability corresponds to the best data association, using which, target states can be estimated. When performed without any approximations, TOMHT is identical to the optimal hypothesis tree, discussed in Figure 1.1 and in Section 4.2.2. The intractability of the optimal solution is handled by making approximations that result in variants of TOMHT, that are not optimal but perform better than the single hypothesis methods. One such strategy based on pruning is the N–scan pruning based MHT, discussed next.

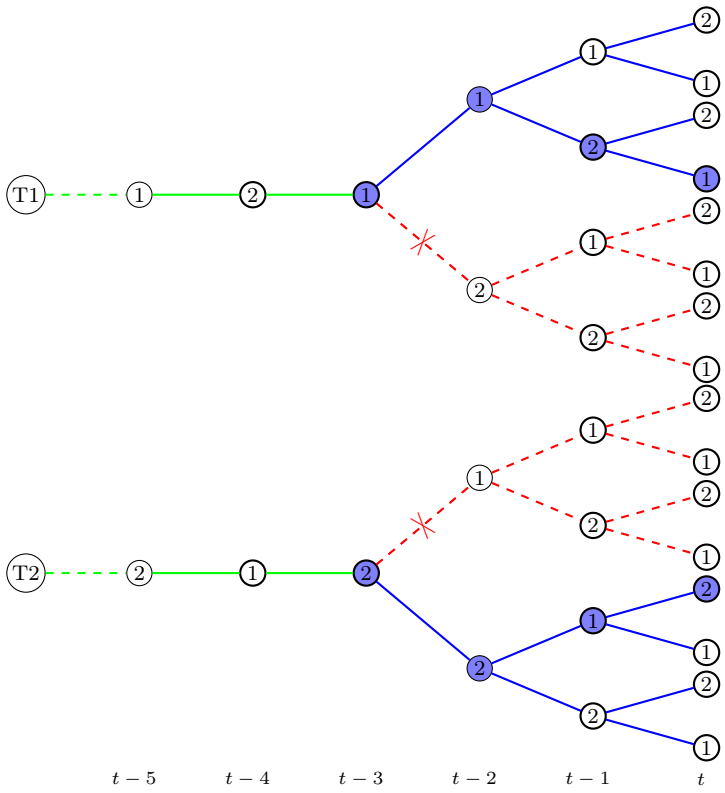


Figure 5.2: *TOMHT with N–scan pruning. Scan depth depicted in the tree is 3. As a result, hard decision is made* $t - -3$*, leaving only a single data association hypothesis at* $t - -3$*, but multiple hypotheses thereafter.*

5.2 N–scan pruning based MHT

The optimal data association solution makes hard decisions on assignments by considering all possible hypotheses, formed using all measurements received at all scans. This has been established to be intractable. One option to deal with this complexity would be to reduce the number of scans of measurements used – an extreme version of this is GNN, which uses measurements just from a single scan. On the other hand, the N–scan pruning TOMHT algorithm [31] makes a trade off between optimality and computational expense. It does not grow hypotheses for all scans of measurements, but only based on N scans of measurements, where $N > 0$. It is evident that this version of MHT becomes GNN when $N = 1$. Scan depths in the range of 3 to 6 are deemed useful, and improve the quality of the tracking solution remarkabbly [20] when compared to single hypothesis methods. The variable N in this version of MHT can be interpreted as the memory of the algorithm in making the data association decisions.

Figure 5.2 shows the N–scan pruning operation on a hypothesis tree for the case of two targets, with scan depth of 3 time scans. At t, there are multiple hypotheses, and the best global hypothesis (assuming hypothesis probabilities are computed) is marked with blue circles at that instant. Using the notion of memory, the N–scan algorithm traverses backwards on the best hypothesis to $(t - 3)$ and fixes the data association that gave rise to the best hypothesis at t, pruning away all other hypothesis (shown in red). Thus, it leaves only a single data association

hypothesis beyond the scan depth, shown as the green links in the figure.

The trade–off in performance and optimality in the N–scan pruning based TOMHT is straightforward. By increasing N, the solution tends towards the optimal solution, but is severely constrained by the computational expense. Depending on the tracking applications and resources at disposal, the value of N can be chosen. The scan depth used in the evaluations presented in this report is for $N = 3$. Even with this depth, TOMHT outperforms all considered algorithms in most scenarios, and hence is used as the benchmark for comparison of both quality of tracks and the computation expense.

There are efficient implementations of the N–scan pruning based TOMHT, where the tree approach can be avoided. These techniques employ the generalized assignment using a sliding window approach, based on Lagrangian relaxation which is discussed in [32].

Chapter 6

Expectation maximization

After having established the basics of multi–target tracking, and the problem of data association with adequate details in the previous chapters, in this chapter the main technique used in the contributions presented in this thesis is discussed. While all methods presented till now attempted to resolve data association uncertainties using pruning or merging of hypotheses in a tree, the solution studied in this thesis avoids working with the notion of an explicit hypothesis tree. Instead, the MAP estimates of target states are approximated using an iterative algorithm – *expectation maximization* (EM). EM is a popular choice for obtaining approximate ML or MAP estimates from mixture models [33], which makes it particularly useful in multi–target tracking, as the true posterior in MTT is usually a mixture model. In this chapter, the EM algorithm is introduced as a method that successively improves the lower bound of an objective function. A discussion relating EM to the broader approximate inference methods – *variational Bayesian* (VB) methods is presented. In multi–target tracking applications, EM has been used in the *probabilistic multiple–hypothesis tracking* (PMHT) algorithm, which is briefly presented, mainly to emphasize the differences between PMHT and the proposed algorithms. First, a discussion on Kullback–Leibler divergence, a measure that is useful in comparing distributions follows.

6.1 Kullback–Leibler divergence

Kullback–Leibler divergence [34] (KLD), is a non–symmetric *measure* of the difference between any two probability distributions[1]. It is a useful metric when approximating complex and unknown distributions with simpler approximating distributions. KLD serves as a good cost function to minimize over, when searching for parameters of the approximating distribution. KLD is denoted as $\mathcal{D}(p||q)$, given as,

$$\mathcal{D}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx, \tag{6.1}$$

where, $p(x)$ is the unknown distribution and $q(x)$ is the approximating distribution[2]. It is common to choose the approximating distributions from simpler, predefined families. In our treatment of KLD in this thesis, we are interested in Gaussian density as the approximating density. From an

[1] For non–trivial comparisons, both distributions must have overlapping support.
[2] This is in fact under the inclusive KLD setting.

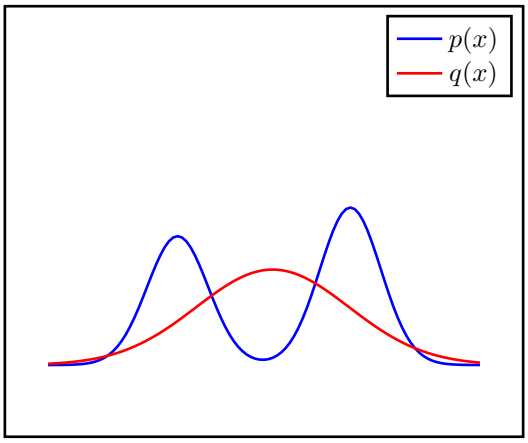


Figure 6.1: *Illustration of a Gaussian density approximating a bimodal distribution, by minimizing the inclusive KLD.*

information theory perspective, KLD captures the *relative entropy* between the two distributions, evident from the form of the expression above [33]. KLD is a positive quantity, vanishing to *zero* only when the two distributions are the same. The non–symmetric nature of KLD allows for the distinction made as the inclusive KLD and exclusive KLD, discussed next.

6.1.1 Inclusive KLD

The approximating distribution, $q(x)$, that minimizes the KLD in (6.1) should have substantial weight at all points where $p(x)$ is non–zero. If the approximating distribution is constrained to be Gaussian, it would imply, $q(x)$ will cover the entire support of $p(x)$. This is illustrated in Figure 6.1. As $q(x)$ tries to *include* all modes of $p(x)$ to the best possible extent with a single Gaussian, this KLD is also referred to as the "inclusive" KLD.

Minimizing the inclusive KLD, by constraining the approximating distribution to Gaussian is equivalent to performing moment matching [24], which was discussed in relation to JPDA in Section 4.4. The notion of merging hypotheses into a single hypothesis is clearly seen in Figure 6.1, where each mode in $p(x)$ can be assumed to be the modes corresponding to two data association hypotheses, wheras the single mode of $q(x)$ corresponds to the single, merged data association hypothesis.

6.1.2 Exclusive KLD

The non–symmetric nature of KLD results in a second measure, minimizing which can yield different approximating distributions. This reverse KLD denoted as, $\mathcal{D}(q||p)$ is given as,

$$\mathcal{D}(q||p) = \int q(x) \log \frac{q(x)}{p(x)} dx. \tag{6.2}$$

Notice that, now the expectation is with respect to the approximating density $q(x)$. From the expression it is clear that, whenever $p(x)$ is vanishing, in order to minimize the KLD in (6.2) $q(x)$ will also be forced to be vanish. This is illustrated for the example of a single Gaussian

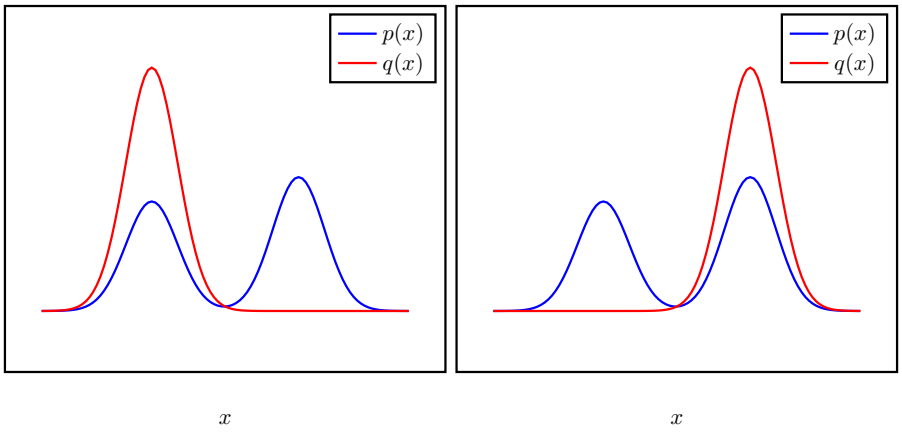


Figure 6.2: *Illustration of two possible approximations to the original bimodal distribution with Gaussians, obtained by minimizing exclusive KLD.*

approximating the bimodal density. When $p(x)$ vanishes to zero, the ratio $q(x)/p(x)$ grows to infinity, and the choice of $q(x)$ that minimizes KLD would exclude either of the modes, as shown in Figure 6.2, thus yielding this version the name "exclusive" KLD. To highlight the fact that based on the initialization, minimizing exclusive KLD can yield different approximating densities Figure 6.2 shows two possible approximating densities for the bimodal density. Many variational inference techniques, including the EM algorithm discussed next, obtain approximating distributions that capture single massive modes by minimizing the exclusive KLD [35].

6.2 Expectation maximization

Parameter estimation turns out to be difficult and intractable, either when partial data is observed, or when the model has some hidden variables in it. It is in such scenarios, that expectation maximization (EM) proves to be most useful. The idea behind EM is highly intuitive – instead of working with incomplete data, EM guesses (loosely speaking) the missing data based on some initial parameters and uses thus obtained complete data to obtain the next set of best parameters. Further, uses these parameters to improve upon the missing data, and so on and so forth. In a more formal sense, in EM for ML estimation, the guess made by EM is the *expectation* of the complete data log–likelihood with respect to an *auxiliary* distribution over the hidden data [36], which is parameterized by an initial set of parameters; this step for obvious reasons is called the E–step. Next, the complete data log–likelihood is *maximized* to obtain new set of parameters, which becomes the M–step. EM iterates between the E and M steps, successively inching towards the MAP or ML estimate. The estimates are ensured to improve with iterations, at least, until the algorithm has not converged to a local maximum [37].

6.2.1 Incomplete data parameter estimation

In Chapter 2, parameter estimation from observed data using ML and MAP estimation methods was discussed; the discussion presented was assuming the observed data was *complete*. The completeness in this context pertains to some missing information, which otherwise makes the parameter estimation intractable. The most common incomplete–data problem is when the model (of which we are estimating the parameters) has hidden variables that are not observed. For

instance, in multi–target tracking, we are interested in the target states, observed via measurements from sensors. But, as we have seen comprehensively in Chapter 4, without measurement–to–target assignment details, target states cannot be estimated easily. In MTT, the data association variables are the hidden variables that are not observed. The complete data for estimation of target states comprises of the sensor measurements and the hidden data association information.

A common occurrence where hidden variables are encountered is when dealing with mixture models. For instance, the problem of parameter estimation in a Gaussian mixture model (Section 2.1.2) can be decomposed into the tasks of estimating parameters of individual Gaussian components, and the mixture weights. The missing information in this case is the relation between mixture components and observed measurements. Given the hidden information, about which component the measurements belong to, the complexity of parameter estimation problem in GMM reduces drastically. This conforms with the discussions in Section 4.2.2, where it was pointed out that the true posterior in MTT is a Gaussian mixture model, with data association variables as the hidden variables.

6.2.2 The EM algorithm for MAP estimation

In this section, derivation of the EM algorithm iterations is presented for the MAP estimation problem, referred to as MAP–EM. EM is widely used for ML estimation, but our interest in this thesis lies in MAP estimation of target states, the derivation presented below focuses directly on MAP–EM. It can be easily shown that EM for ML estimation is a special case of the MAP–EM, when the prior is non–informative (uniform). The following derivation is carried out using notations common in EM literature.

If $\mathbf{Y}$ is the incomplete data that is observed, $\mathbf{Z}$ is the discrete *hidden* variable, then the MAP estimate of parameters $\boldsymbol{\theta}$ is given as,

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{Y}) \tag{6.3}$$

$$\equiv \arg\max_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta}, \mathbf{Y}) \tag{6.4}$$

Introducing natural logarithm over the joint density does not alter the MAP estimation, as logarithm is a monotonically increasing function; this equivalence is discussed in context of ML estimation with log–likelihood functions in Section 2.2.1. It can be noticed that the joint density $p(\boldsymbol{\theta}, \mathbf{Y})$ is based on the incomplete data, meaning, the MAP estimation problem is as yet intractable. Next, the hidden variables are introduced to obtain the density with complete data,

$$\ln p(\boldsymbol{\theta}, \mathbf{Y}) = \ln \sum_{\mathbf{Z}} p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{Y}). \tag{6.5}$$

In the last step, hidden variables, $\mathbf{Z}$, are introduced using the law of total probability; if $\mathbf{Z}$ is continuous the summation is replaced by integral and the expression still holds. Although the data is complete with the hidden variables, MAP estimation over all possible values of the hidden variables in (6.5) is intractable; in fact it is NP–hard [38]. Further, the expression in the last step is of log–of–sum form, which is difficult to handle. To tackle this intractable log–sum objective function, a distribution over the hidden variables is introduced, $q_{\mathbf{Z}}(\mathbf{Z})$, and the logarithm is

pushed inside the summation using Jensen's inequality[3],

$$\ln p(\boldsymbol{\theta}, \mathbf{Y}) = \ln \sum_{\mathbf{Z}} q_{\mathbf{Z}}(\mathbf{Z}) \frac{p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{Y})}{q_{\mathbf{Z}}(\mathbf{Z})}. \tag{6.6}$$

$$\geq \sum_{\mathbf{Z}} q_{\mathbf{Z}}(\mathbf{Z}) \ln \frac{p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{Y})}{q_{\mathbf{Z}}(\mathbf{Z})} \triangleq \mathcal{F}(q_{\mathbf{Z}(\mathbf{Z})}, \boldsymbol{\theta}). \tag{6.7}$$

The term $\mathcal{F}(q_{\mathbf{Z}(\mathbf{Z})}, \boldsymbol{\theta})$ can be interpreted as the lower bound on the log–joint density, $\ln p(\boldsymbol{\theta}, \mathbf{Y})$.

The difference between the true log–joint density and the lower bound, yields the exclusive KLD between $q_{\mathbf{Z}}(\mathbf{Z})$ and $\Pr\{\mathbf{Z}|\boldsymbol{\theta}, \mathbf{Y}\}$, which can be seen by rewriting the lower bound as,

$$\mathcal{F}(q_{\mathbf{Z}(\mathbf{Z})}, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q_{\mathbf{Z}}(\mathbf{Z}) \ln \frac{\Pr\{\mathbf{Z}|\boldsymbol{\theta}, \mathbf{Y}\} p(\boldsymbol{\theta}, \mathbf{Y})}{q_{\mathbf{Z}}(\mathbf{Z})} \tag{6.8}$$

$$= \sum_{\mathbf{Z}} q_{\mathbf{Z}}(\mathbf{Z}) \ln \frac{\Pr\{\mathbf{Z}|\boldsymbol{\theta}, \mathbf{Y}\}}{q_{\mathbf{Z}}(\mathbf{Z})} + \sum_{\mathbf{Z}} q_{\mathbf{Z}}(\mathbf{Z}) \ln p(\boldsymbol{\theta}, \mathbf{Y}) \tag{6.9}$$

As, $q_{\mathbf{Z}}(\mathbf{Z})$ is a probability distribution, $\sum_{\mathbf{Z}} q_{\mathbf{Z}}(\mathbf{Z}) = 1$,

$$\ln p(\boldsymbol{\theta}, \mathbf{Y}) = \mathcal{F}(q_{\mathbf{Z}(\mathbf{Z})}, \boldsymbol{\theta}) + \mathcal{D}(q||p), \tag{6.10}$$

where $\mathcal{D}(q||p)$ is the exclusive KLD between $q_{\mathbf{Z}}(\mathbf{Z})$ and $\Pr\{\mathbf{Z}|\boldsymbol{\theta}, \mathbf{Y}\}$ defined in (6.2). As $\mathcal{D}(q||p) \geq 0$, $\mathcal{F}(q_{\mathbf{Z}(\mathbf{Z})}, \boldsymbol{\theta})$ is indeed the lower bound. It is easy to see from (6.10) that maximizing the lower bound is equivalent to minimizing the KLD. In fact, when $\mathcal{D}(q||p) = 0$, the lower bound is exact! While it is elegant to have a lower bound of the form in (6.7), it is much more useful when it is a tight bound. EM iteratively maximizes the lower bound, and in this process it also approximates the MAP estimate. This is shown next.

The lower bound in (6.7) is a functional of the free distribution $q_{\mathbf{Z}}(\mathbf{Z})$ and parameters $\boldsymbol{\theta}$. By iteratively maximizing each of $q_{\mathbf{Z}}(\mathbf{Z})$ and $\boldsymbol{\theta}$, when holding the other fixed, the lower bound can be maximized. These iterations form the basis for EM, given below, with n as the iteration index.

E–step:

$$q_{\mathbf{Z}}^{(n)}(\mathbf{Z}) = \arg\max_{q_{\mathbf{Z}}(\mathbf{Z})} \mathcal{F}(q_{\mathbf{Z}}(\mathbf{Z}), \boldsymbol{\theta}^{(n)}) \tag{6.11}$$

M–step:

$$\boldsymbol{\theta}^{(n+1)} = \arg\max_{\boldsymbol{\theta}} \mathcal{F}(q_{\mathbf{Z}}^{(n)}(\mathbf{Z}), \boldsymbol{\theta}) \tag{6.12}$$

EM iterations are guaranteed to increase this lower bound, with each iteration, until a maximum is attained [39]. For a fixed parameter setting, $\boldsymbol{\theta}^{(n)}$, the distribution that minimizes $\mathcal{D}(q||p)$ can be found out simply by inspecting the KLD,

$$\mathcal{D}(q||p) = \sum_{\mathbf{Z}} q_{\mathbf{Z}}(\mathbf{Z}) \ln \frac{q_{\mathbf{Z}}(\mathbf{Z})}{\Pr\{\mathbf{Z}|\boldsymbol{\theta}^{(n)}, \mathbf{Y}\}}. \tag{6.13}$$

The KLD is zero, and the lower bound becomes exact when $q_{\mathbf{Z}}^{(n)}(\mathbf{Z}) = \Pr\{\mathbf{Z}|\boldsymbol{\theta}^{(n)}, \mathbf{Y}\}$, thus, the (6.11) becomes,

E–step:

$$q_{\mathbf{Z}}^{(n)}(\mathbf{Z}) = \Pr\{\mathbf{Z}|\boldsymbol{\theta}^{(n)}, \mathbf{Y}\}. \tag{6.14}$$

[3]Jensen's inequality states that, for any concave function g and the expectation operation $\mathbf{E}$ over some distribution of variable X,

$$g\mathbf{E}(X) \geq \mathbf{E}(g(X)).$$

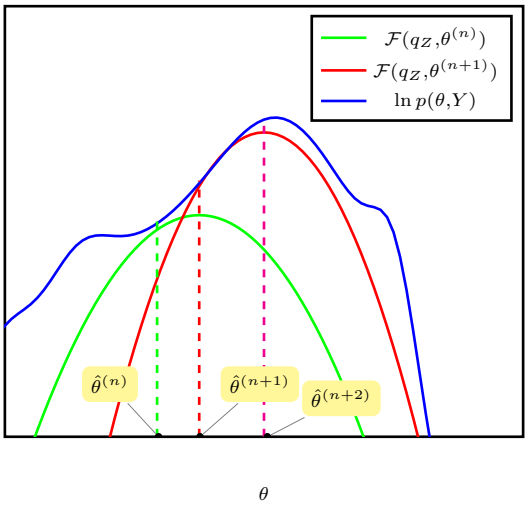


Figure 6.3: *Successive maximization of the lower bound, in EM iterations.*

Using the distribution obtained in the E–step, parameters are maximized, yielding,

$$\boldsymbol{\theta}^{(n+1)} = \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{Z}} \Pr\{\mathbf{Z}|\boldsymbol{\theta}^{(n)}, \mathbf{Y}\} \ln \frac{p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{Y})}{\Pr\{\mathbf{Z}|\boldsymbol{\theta}^{(n)}, \mathbf{Y}\}}. \tag{6.15}$$

As the maximization is over the parameters, the factor in the denominator of the objective function does not contribute to the MAP estimation, thus,
M–step:

$$\boldsymbol{\theta}^{(n+1)} = \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{Z}} \Pr\{\mathbf{Z}|\boldsymbol{\theta}^{(n)}, \mathbf{Y}\} \ln p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{Y}). \tag{6.16}$$

Equations (6.14) and (6.16) form the E and the M steps of EM algorithm, respectively, and are iteratively computed until convergence.

Figure 6.3[4] depicts iterations of EM as a means to increase the lower bound on the log–joint density, $\ln p(\boldsymbol{\theta}, \mathbf{Y})$. At iteration n, the parameters yield a lower bound, which is then maximized to obtain the next set of parameters. It can also be noted that for a given parameter setting the lower bound touches the true objective function i.e, the bound is exact. When the new lower bound is computed using these updated parameters, it is larger than the previous bound. With each iteration, the MAP estimates improve approaching the true global maximum.

6.3 EM in target tracking

The objective in MTT is to obtain MAP estimate of target states from the true posterior density, which is a mixture–model. As mixture models can be treated as models with hidden parameters, the EM algorithm can be conceptually applied to approximate the estimates. One of the earlier attempts at using EM to MTT was in probabilistic multiple hypothesis tracking [40] (PMHT), which is briefly presented next. PMHT is also of interest, as both the algorithms presented in this thesis are based on EM, and the algorithm presented in Paper A is close to PMHT.

[4]Based on Fig.9.14 in [33]

6.3.1 Probabilistic multiple hypothesis tracking

In Chapter 4, single hypothesis methods like GNN and JPDA were presented; both the algorithms use measurements from a single scan in forming the single global hypothesis. While MHT discussed in Chapter 5 uses multiple scans of data, it is severely constrained by the computational complexity. PMHT on the other hand, is a batch algorithm: it uses measurements from all time instants. It is a linear–complexity algorithm which uses EM to iteratively estimate target states, by treating the data association variable as the hidden variable.

PMHT is similar to JPDA which was discussed in Section 4.4. Being a filtering solution, JPDA used measurements from a single scan to make its data association decisions. It was demonstrated that JPDA computes marginal data association probabilities, to form composite measurements ensuring the point target constraints are satisfied. PMHT uses the notion of composite measurements to update the targets' predicted states. But, to reduce complexity it does not use marginal data association probabilities as the weights. PMHT does not take into consideration the conditioning due to a measurement being associated another target. This implies, PMHT does not satisfy the point target model assumptions and allows for possibly the case when many measurements are generated from a single target. This is the main difference between PMHT and the algorithm presented in Paper A.

References

[1] J. Hsu. (2014). Google's self-driving car prototype ready to hit the road, [Online]. Available: http://spectrum.ieee.org/cars-that-think/transportation/self-driving/googles-selfdriving-car-prototype-ready-to-hit-the-road.

[2] E. Ackerman. (2015). Spot is Boston Dynamics' nimble new quadruped robot, [Online]. Available: http://spectrum.ieee.org/automaton/robotics/robotics-hardware/spot-is-boston-dynamics-nimble-new-quadruped-robot.

[3] ——, (May 2015). Automation in cars: a $100 billion market by 2030, [Online]. Available: http://spectrum.ieee.org/cars-that-think/transportation/advanced-cars/automation-in-cars-a-100-billion-market-by-2030.

[4] A. Davies. (May 2015). How daimler built the world's first self-driving semi, [Online]. Available: http://www.wired.com/2015/05/daimler-built-worlds-first-self-driving-semi/.

[5] R. L. Mitchell, Models of extended targets and their coherent radar images, *Proceedings of the IEEE* vol. **62**, no. 6 1974, 754–758, 1974.

[6] P. Diaconis, D. Ylvisaker, *et al.*, Conjugate priors for exponential families, *The Annals of statistics* vol. **7**, no. 2 1979, 269–281, 1979.

[7] C. M. Bishop *et al.*, "Pattern recognition and machine learning", 4. springer New York, 2006, vol. 4, ch. 2,9.

[8] M. K. Steven, "Fundamentals of statistical signal processing: Estimation theory". Prentice Hall Signal Processing Series, AV Oppenheim, Ed. Prentice Hall PTR, 1993, vol. 1, ch. 7.

[9] S. Boyd and L. Vandenberghe, "Convex optimization". Cambridge university press, 2004, ch. 3.

[10] D. Koller and N. Friedman, "Probabilistic graphical models: principles and techniques". MIT press, 2009, ch. 17.

[11] M. K. Steven, "Fundamentals of statistical signal processing: Estimation theory". Prentice Hall Signal Processing Series, AV Oppenheim, Ed. Prentice Hall PTR, 1993, vol. 1, ch. 10,11.

[12] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.

[13] C. M. Bishop *et al.*, "Pattern recognition and machine learning", 4. springer New York, 2006, vol. 4, ch. 13.

[14] R. E. Kalman, A new approach to linear filtering and prediction problems, *Journal of Fluids Engineering* vol. **82**, no. 1 1960, 35–45, 1960.

[15] G. Kitagawa, Non-gaussian state—space modeling of nonstationary time series, *Journal of the American statistical association* vol. **82**, no. 400 1987, 1032–1041, 1987.

[16] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood: Artech House, 1999.

[17] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: modeling social behavior for multi-target tracking", *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009, pp. 261–268.

[18] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[19] B. T. Vo, "Random finite sets in multi-object filtering", PhD thesis, University of Western Australia, 2008.

[20] S. S. Blackman, Multiple-target tracking with radar applications, *Dedham, MA, Artech House, Inc., 1986, 463 p.* vol. **1** 1986, 1986.

[21] D. P. Bertsekas, The auction algorithm: A distributed relaxation method for the assignment problem, *Annals of operations research* vol. **14**, no. 1 1988, 105–123, 1988.

[22] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, *IEEE Journal of Oceanic Engineering* vol. **8**, no. 3 1983, 173–184, 1983.

[23] D. Svensson, "Target tracking in complex scenarios", PhD thesis, Chalmers University of Technology, 2010.

[24] A. R. Runnalls, Kullback-leibler approach to gaussian mixture reduction, *Aerospace and Electronic Systems, IEEE Transactions on* vol. **43**, no. 3 2007, 989–999, 2007.

[25] J. L. Williams and R. A. Lau, "Data association by loopy belief propagation", *International Conference on Information Fusion*, IEEE, 2010, pp. 1–8.

[26] H. Lang and C. Shan, Bias phenomenon and compensation in multiple target tracking algorithms, *Mathematical and Computer Modelling* vol. **31**, no. 8 2000, 147–165, 2000.

[27] R. Fitzgerald, Track biases and coalescence with probabilistic data association, *Aerospace and Electronic Systems, IEEE Transactions on* no. 6 1985, 822–825, 1985.

[28] D. B. Reid, An algorithm for tracking multiple targets, *IEEE Transactions on Automatic Control* vol. **24**, no. 6 1979, 843–854, 1979.

[29] S. S. Blackman, Multiple hypothesis tracking for multiple target tracking, *Aerospace and Electronic Systems Magazine, IEEE* vol. **19**, no. 1 2004, 5–18, 2004.

[30] K. G. Murty, An algorithm for ranking all the assignments in order of increasing cost, *Operations Research* vol. **16**, no. 3 1968, 682–687, 1968.

[31] T. Kurien, M. Liggins, *et al.*, "Report-to-target assignment in multisensor multitarget tracking", *IEEE Conference on Decision and Control*, IEEE, 1988, pp. 2484–2488.

[32] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, A generalized SD assignment algorithm for multisensor-multitarget state estimation, *IEEE Transactions on Aerospace and Electronic Systems* vol. **33**, no. 2 1997, 523–538, 1997.

[33] C. M. Bishop *et al.*, *Pattern recognition and machine learning*, 4. springer New York, 2006, vol. 4.

[34] S. Kullback and R. A. Leibler, On information and sufficiency, *The annals of mathematical statistics* 1951, 79–86, 1951.

[35] B. J. Frey, R. Patrascu, T. S. Jaakkola, and J. Moran, Sequentially fitting" inclusive" trees for inference in noisy-or networks 2000, 2000.

[36] M. J. Beal, "Variational algorithms for approximate bayesian inference", PhD thesis, University of London, 2003.

[37] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[38] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[39] C. J. Wu, On the convergence properties of the EM algorithm, *The Annals of statistics* 1983, 95–103, 1983.

[40] R. L. Streit and T. E. Luginbuhl, "Probabilistic multi-hypothesis tracking", DTIC Document, Tech. Rep., 1995.

Part II

Publications

Paper A

Tracking multiple point targets using expectation maximization

A.S.Rahmathullah, R. Selvan and L. Svensson

Submitted to IEEE Transactions on Signal Processing

Tracking multiple point targets using expectation maximization

Abu Sajana Rahmathullah, Raghavendra Selvan, Lennart Svensson

Abstract

In this paper, we propose a strategy that is based on expectation maximization for tracking multiple point targets. The algorithm is similar to probabilistic multi-hypothesis tracking (PMHT), but does not relax the point target model assumptions. According to the point target models, a target can generate at most one measurement and a measurement is generated by at most one target. With this model assumption, we show that the proposed algorithm can be implemented as iterations of Rauch-Tung-Striebel (RTS) smoothing for state estimation, and the loopy belief propagation method for marginal probabilities calculation. Using example illustrations with tracks, we compare the proposed algorithm with PMHT and joint probabilistic data association (JPDA) and show that PMHT and JPDA exhibit coalescence when there are closely moving targets whereas the proposed algorithm does not. Furthermore, extensive simulations comparing the mean optimal sub-pattern assignment (MOSPA) performance of the algorithm for different scenarios averaged over several Monte Carlo iterations show that the proposed algorithm performs better than JPDA and PMHT. We also compare it to benchmarking algorithm: N-scan pruning based track-oriented multiple hypothesis tracking (TOMHT). The proposed algorithm shows a good trade-off between computational complexity and the MOSPA performance.

1 Introduction

There are many interesting applications based on tracking multiple targets, and the range of applications are only increasing by the day. With the well justified enthusiasm in the domain of autonomous vehicular navigation [20], computer vision [17] and the classical radar based tracking applications [2], the need for tractable multi-target tracking algorithms that perform close to optimal in real-time, has risen in the last few years. The problem of multi-target tracking is widely studied [6, 21, 26], and newer paradigms are being investigated to solve existing bottlenecks, and to enable further possibilities [16].

The focus of this paper is on tracking multiple point targets, and one of the main challenges in such a tracking problem is to resolve measurement uncertainties [1, 30]. Discerning target-generated measurements from clutter, and finding correspondence between targets and measurements, in order to estimate target tracks in space and time forms the crux of the problem. The uncertainty in data association can be handled by maintaining data association hypotheses, which map partitions of measurement data to different targets [3, 5]. While this approach of maintaining data hypotheses might seem plausible to resolve uncertainties, it scales poorly with number of targets and measurements, when aggregated over different time instants [19]. The number of hypotheses grows exponentially, and most existing multi-target tracking algorithms propose solutions which can circumvent handling massive number of data association hypotheses, making a trade-off between optimality and complexity [28].

One of the earliest multi-target tracking algorithms developed was multiple hypothesis tracking (MHT) [5,6,30], with a possibility of maintaining all data association hypotheses in a tree form, mak-

ing it optimal and at the same time rendering it computationally intractable. To reduce complexity there have been many approximations to the original MHT, that do not maintain all possibilities in the hypothesis tree . One such common version of MHT is the N-scan pruning based track-oriented MHT [19], which maintains a hypothesis tree only for the latest N scans of measurements, and prunes away all hypothesis before N scans, in effect leaving only a single data association hypothesis before N scans. While this strategy reduces complexity, low scan depths can considerably degrade performance of the algorithm.

Another strategy that has received considerable attention in the past is the joint probabilistic data association (JPDA) filter [5, 8, 12]. JPDA performs moment matching at each time instant to merge multiple hypotheses into a single hypothesis, which is equivalent to performing the filter update with composite measurements for linear models. The composite measurements are obtained by weighting measurements at each time instant with their corresponding marginal data association probabilities. Computing marginal probabilities for a high number of targets can be expensive, but there exist efficient approximate techniques like k-best hypotheses based on Murty's algorithm [5,24], Monte Carlo sampling method [23], tree-based approach [15] and loopy belief propagation (LBP) [37] algorithm. Another disadvantage with JPDA is coalescence which is a consequence of moment matching when there are closely moving targets [7, 9].

Probabilistic multiple hypothesis tracking (PMHT) [32,35], on the other hand, uses multiple scans of data and iteratively optimizes the maximum a posteriori (MAP) estimate of the target states, using the expectation maximization (EM) algorithm [11, 33]. While PMHT handles measurement data from all scans in its EM iterations, it does not maintain a hypothesis tree, resulting in reduced complexity when compared to MHT. Further, PMHT makes a model assumption that befits extended targets and not point targets [10], to avoid the computation of marginal data association probabilities. This relaxation further reduces complexity, but makes PMHT susceptible to track coalescence.

In this paper, we propose a batch algorithm that uses EM [33] to obtain MAP estimates of target states, by treating the data association variable as a hidden variable, for the case of known number of targets. We show that the main steps involved in the algorithm are smoothing and computing marginal data association probabilities. It turns out that the algorithm we propose is similar to PMHT, as both are batch solutions and use EM to obtain MAP estimate of target states, but with an important difference in the model assumptions: our assumptions are consistent for point targets, meaning, each target is assumed to generate and hence be associated only to one measurement, whereas this constraint is relaxed in PMHT. Further, the need to compute marginal data association probabilities in the proposed algorithm is similar to JPDA, but differing in the fact that we do not perform moment matching. Our implementation of the algorithm uses Rauch-Tung-Striebel (RTS) smoother [29] for smoothing, and an extension to the LBP [37] to approximate the marginal probabilities. Although it is beyond the scope of the current article, the strategies used to turn PMHT into an online algorithm and to perform track-handling can be adopted to extend our algorithm on the same lines.

We compare the performance of the proposed algorithm with MHT, PMHT and JPDA. We illustrate the conceptual differences with PMHT and JPDA, while comparing it to the benchmark algorithm, the N-scan pruning-based TOMHT algorithm. We show that the proposed algorithm does not suffer from track coalescence, when there are closely moving or crossing targets, in contrast to the PMHT or the JPDA algorithms. Plus, a good initialisation can significantly improve the performance of the proposed algorithm compared to PMHT. We also compare these algorithms for different scenarios to highlight the accuracy and computational advantage of the proposed algorithm. In all scenarios considered, the proposed algorithm shows improvements in mean optimal sub-pattern assignment (MOSPA) [14, 31] metric when compared to PMHT and JPDA, while being computationally cheaper than MHT.

The organization of the paper is as follows: In Section 2, we present the model assumptions and main challenges addressed in this paper. In Sections 3 and 4, our main contributions are presented,

where we elaborate on the role of expectation maximization and computing marginal probabilities in the presented algorithm. In Section 5, algorithm-level details of the proposed solution are presented. In Section 6, the proposed solution is contrasted with two closely related methods, PMHT and JPDA. Performance evaluation of our solution is presented in Section 7. Finally, we present our conclusions and possible tracks for future work in Section 8.

2 Problem formulation and background

In this section, we begin by presenting the model assumptions based on which the algorithm will be derived, followed by a discussion on the problem we are interested in solving. After that, we describe a few existing multi-target tracking algorithms, providing a context for our contribution.

2.1 Problem formulation

We consider multiple point targets moving in a cluttered background. The number of targets, denoted N_T, is assumed to be known and all targets are assumed to be present in the observation region at all times. The state vector $X_{t,i}$ for target i at time t is varying according to the process model,

$$X_{t,i} = FX_{t-1,i} + V_{t,i}, \tag{1}$$

where $V_{t,i} \sim \mathcal{N}(0, Q)$. This model implies that targets move independently of other targets. We also assume we have a Gaussian prior distribution on each target i with mean $\mu_{0,i}$ and covariance $P_{0,i}$. A target is detected with probability P_D and the target when detected gives a measurement according to

$$Y_{t,i}^{\text{target}} = HX_{t,i} + W_{t,i}, \tag{2}$$

where $W_{t,i} \sim \mathcal{N}(0, R)$. Thus, the measurement set Y_t at each time is the union of the target measurements (of the detected targets) and a set of clutter detections. The clutter measurements are assumed to be distributed according to a Poisson process with intensity $\beta_c V$ where V is the volume of the observation region and β_c is the clutter density per volume. The number of measurements obtained at time t is denoted M_t.

In this paper, we consider a batch problem in which we have access to the data $Y = (Y_1, \ldots, Y_T)$ and we seek to obtain the MAP estimate of the state $X = (X_{1,1}, \ldots, X_{1,N_T}, X_{2,1}, \ldots, X_{T,N_T})$, given as

$$X_{\text{MAP}} \quad = \quad \arg\max_X p(X|Y). \tag{3}$$

That is, the goal is to estimate the trajectories of all targets. Also, an estimate of the quality of the state estimates, often captured in a covariance matrix, is desirable.

The density involved in MAP estimation is $p(X|Y)$, and evaluating this density is tractable and straightforward if there is no uncertainty in the measurement origin,for the model assumptions described above. However, in the multi-target tracking problems, the measurement set comprises both the measurements from the targets that are detected and the clutter measurements. And the information regarding which measurements correspond to which targets is not available. To describe this uncertainty in the measurement origin, one traditional way is to introduce the data association variable $K = (k_{1,1}, \ldots, k_{T,N_T})$, where $k_{t,i} = j$ denotes assigning the target i at time t to the measurement $Y_{t,j}$. Note that these variables should take values that satisfy the point target assumptions, i.e., a target can be assigned to at most one measurement and a measurement can be assigned to at most one target. With these variables, the density of interest becomes $p(X, K|Y)$, using which the estimates can be computed. For instance, consider $X_{\text{MAP}} = \arg\max_X \sum_K p(X, K|Y)$ for MAP estimation of X. Here, introducing the data association variable makes it easier to represent

the measurement uncertainty. However, the estimation problem is still intractable due to the sheer number of possibilities of K. The number of possible hypotheses, K, grows exponentially with the number of measurements and the trajectory length, thus making it intractable to marginalize or maximize with respect to K. In the remainder of this section, we will give a brief overview of some of the existing sub-optimal algorithms to estimate X. Adhering to the conventional terminology, we will refer to an instance of K as the data association hypothesis.

2.2 Background

For each K, the density $p(X|K,Y)$ is a Gaussian density based on our model assumptions. This implies that $p(X|Y)$ is a Gaussian mixture (GM) with exponential number of terms. Handling such mixture densities is prohibitive, and some of the sub-optimal multi-target tracking algorithms resort to estimating X by making approximations to the posterior density. In the remainder of this section, we will give a summary of a few existing algorithms, the approximations involved and their limitations. Two of the algorithms discussed here, namely, MHT and JPDA, are for filtering problems, wherein the algorithms obtain target states X_t recursively by approximating the posterior density $p(X_t|Y_{1:t})$ every time instant, where $Y_{1:t}$ denotes the set of measurements from time 1 to t. It is possible to extend these filtering algorithms to batch problems by performing smoothing on the filtered estimates. We also discuss a batch solution, PMHT, which is based on EM and is very similar to the proposed algorithm in this paper.

The JPDA algorithm, first introduced in [12], approximates the posterior density GM $p(X_t|Y_{1:t})$ as a single Gaussian using moment matching. For linear models, moment matching of the posterior density is equivalent to updating the prediction density with a single composite measurement [5] $\tilde{Y}_{t,i}$,

$$\tilde{Y}_{t,i} = \sum_{j=1}^{M_t} \omega_{t,i,j} Y_{t,j}. \tag{4}$$

Here, $\omega_{t,i,j}$, is the marginal probability that measurement $Y_{t,j}$ is associated to target i, given by,

$$\omega_{t,i,j} = \sum_{K:k_{t,i}=j} \Pr\{K_{1:t}|Y_{1:t}\}. \tag{5}$$

For linear-Gaussian point target models, (5) can be written as,

$$\omega_{t,i,j} \propto \sum_{K:k_{t,i}=j} \prod_{t,i} \left(\frac{P_D \mathcal{N}(Y_{t,j}; H\mu^p_{t,i}, HP^p_{t,i}H' + R)}{(1-P_D)\beta_c} \right)^{\mathbb{1}_{k_{t,i}}}, \tag{6}$$

where $\mu^p_{t,i}$ and $P^p_{t,i}$ denote the mean and covariance of the predicted density at time t for target i, and and the indicator function $\mathbb{1}_{k_{t,i}}$, is described as, $\mathbb{1}_{k_{t,i}} = \begin{cases} 1 & k_{t,i} \neq 0 \\ 0 & k_{t,i} = 0 \end{cases}$. As can be observed in (4), we need to know the marginal probabilities $\omega_{t,i,j}$ for the measurement update. A straightforward computation of these marginal probabilities is expensive, again because of the number of possibilities of K with $k_{t,i} = j$ and also due to the constraints on K for point targets. That is, for each $w_{t,i,j}$, all the possible hypotheses K that assigns target i to measurement j should be considered; in addition, these hypotheses should ensure point target assumptions and be such that a target is assigned to at most one measurement and a measurement is assigned to at most one target. In the literature, there are different strategies to compute these marginal probabilities by efficiently traversing through the hypotheses [15, 23, 36]. The methods proposed in [23] and [36] are approximations whereas the method in [15] returns the exact marginal probabilities, trading off complexity for accuracy.

A disadvantage of moment matching in JPDA is that it suffers from track coalescence problem when there are closely moving targets. In such scenarios, the priors and likelihoods are almost identical for the two targets. As a result, the posterior, which is a GM can have equally likely Gaussian components, which when moment matched yields almost identical tracks for the targets leading to coalescence [7, 9]. Note that this problem is also possible in many of the tracking algorithms [13], however it is prevalent in the JPDA case.

Like JPDA, MHT is also a filtering algorithm but maintains many Gaussian components in the posterior $p(X_t|K_{1:t})$. In other words, it maintains many hypotheses $K_{1:t}$ at each time instant. Before propagation to the next time, it prunes away hypotheses $K_{1:t}$ that are insignificant, which corresponds to reducing the number of global hypotheses in MHT nomenclature. For this pruning purpose, N-scan pruning [5, 6, 30] is one of the commonly implemented algorithms in MHT. The N-scan pruning algorithm traces back N time scans and prunes away all hypotheses at $t - N$, but the one that is the ancestor of the current best hypothesis $K^*_{1:t}$. Thus, the history before $t - N$ has just one data association possibility. The accuracy of the MHT algorithm depends on the depth N. Larger the N is, the better the accuracy is. However, larger N implies that at each t there will be many hypotheses retained which in turn implies higher computational complexity.

In case of PMHT algorithms [32], the MAP estimates of the states are obtained using a variational inference method, expectation maximization. This batch method is iterative and involves several local optimizations,

$$X^{(n+1)} = \arg\max_X \sum_K \Pr\{K|Y, X^{(n)}\} \ln p(X, K, Y) \tag{7}$$

where the superscripts (n) and $(n + 1)$ refer to the iteration indices. As one can observe from the above equation, there is the distribution $\Pr\{K|Y, X^{(n)}\}$ on K. With each iteration, this distribution on K varies and the state is estimated according to this new distribution. One important aspect regarding PMHT is that it allows a target to be associated to multiple measurements, which is more suitable for the extended target models than the point target model assumptions presented in the beginning of this section. As a consequence, PMHT allows the same measurement to be associated to multiple targets, which can lead to track coalescence when there are closely moving targets. In the next section we present an algorithm for tracking of multiple point targets which is based on EM similar to PMHT. But in contrast to PMHT, we derive the algorithm without relaxing the point target assumptions.

3 EM for target tracking

In this section, we apply EM to a batch version of multiple point target tracking problem and derive our solution. We show that E and M steps in the proposed algorithm mainly comprise of obtaining marginal data association probabilities, which serve as weights for the composite measurements, and smoothing using thus obtained composite measurements. We first present a background on EM and continue with the derivation for the model assumptions in Section 2.

3.1 Expectation maximization background

Expectation maximization (EM), first discussed in [11], is an iterative technique, widely used to obtain approximate ML or MAP estimates of parameters from incomplete observed data. Equivalently, EM is a common choice for parameter estimation when the model has hidden variables, relating the data with parameters. We present a brief introduction to EM for MAP estimation, touching upon the idea of it being a strategy that increases a lower bound on the logarithm of the joint density.

The MAP estimation of state variable X, including the hidden data association variable can be written as,

$$\arg\max_X p(X,Y) \quad = \quad \arg\max_X \ln \sum_K p(X,Y,K). \tag{8}$$

The logarithm of summation over K in (8) has exponential complexity, making the MAP estimation problem NP-hard. EM tackles this complexity of working through all possible values of K by introducing a distribution over the data association variable, $q_K(K)$, and obtaining a term that can be interpreted as a lower bound on $\ln p(X,Y)$ [4, 11], shown below using Jensen's inequality,

$$\ln p(X,Y) \geq \sum_K q_K(K) \ln \frac{p(X,Y,K)}{q_K(K)}. \tag{9}$$

It is evident from (9) that the lower bound is a functional of q_K and X. Using them as parameters in the n-th iteration, EM alternates between finding the best estimate $q_K^{(n)}(K)$ given $X^{(n)}$, and finding the MAP estimate $X^{(n+1)}$ given $q_K^{(n)}(K)$. This alternative optimization is guaranteed to increase the lower bound $\ln p(X,Y)$ until convergence to an optimum [11], at which point the MAP estimate obtained would be the best approximation. Thus, using the lower bound (9) in (8), and collecting the terms dependent on X, the original MAP problem can now be written as,

$$\arg\max_X p(X,Y) \approx \arg\max_{X,q_K} \sum_K q_K(K) \ln p(X,Y,K). \tag{10}$$

From the above equation it is clear that there is an expectation (E-step) with respect to $q_K(K)$ and a maximization (M-step) with respect to X, which form the core of the EM algorithm iterations. It can be shown that the E and M step iterations can be uncoupled as [4, 22],

$$q_K^{(n)}(K) \quad = \quad \Pr\{K|X^{(n)},Y\} \tag{11}$$

$$X^{(n+1)} \quad = \quad \arg\max_X \sum_K q_K^{(n)}(K) \ln p(X,Y,K). \tag{12}$$

The main computational advantage of using EM is that it converts the log-sum problem in (8), into the sum-log form in (12), as this gain can be often substantial. Further, in cases where the joint distribution can be expressed as a product of distributions from the exponential family, these factors reduce to polynomial terms, which can further decrease the complexity.

3.2 Derivation of the EM algorithm for tracking multiple point targets

In this subsection, we derive the E and M steps for the point target tracking problem formulated in Section 2. The E-step derivation is immediately obtained from the model description, whereas, we derive the M-step in some detail to show that the MAP estimate of states can be obtained by smoothing using composite measurements. We show that the composite measurements are weighted by the marginal probabilities of the data association variables, that are in turn distributed according to the density q_K returned by the E-step.

Starting with the expression for the E-step from (11),

$$q_K^{(n)}(K) \quad = \quad \Pr\{K|X^{(n)},Y\} \tag{13}$$

$$\quad \propto \quad p(K,Y|X^{(n)}). \tag{14}$$

For the Gaussian model assumptions in Section 2, it can be shown that [5, 35]

$$p(K, Y|X) \propto \prod_{i,t} p(Y_{t,j}, k_{t,i} = j|X_{t,i}) \tag{15}$$

where,

$$p(Y_{t,j}, k_{t,i} = j|X_{t,i}) = \begin{cases} (1 - P_D)\beta_c & k_{t,i} = 0 \\ P_D \mathcal{N}(Y_{t,j}; HX_{t,i}, R) & k_{t,i} \neq 0 \end{cases}.$$

Incorporating the conditions described above using an indicator function $\mathbb{1}_{k_{t,i}}$, it follows that

$$q_K^{(n)}(K) \propto \prod_{t,i} \left(\frac{P_D \mathcal{N}(Y_{t,k_{t,i}}; HX_{t,i}^{(n)}, R)}{(1 - P_D)\beta_c} \right)^{\mathbb{1}_{k_{t,i}}}, \tag{16}$$

where $X_{t,i}^{(n)}$ is the state estimate target i at time t.

In the M-step, we solve a convex maximization problem at iteration n. There are many ways to solve such problems, but our strategy is to show that the criterion can be rewritten as a Gaussian distribution for which the mean and the covariance have closed form expressions. Based on that result, it is trivial to see that the desired MAP estimate is the mean of that Gaussian distribution.

We begin by rewriting (12) as,

$$X^{(n+1)} = \arg\max_X \exp\left(\sum_K q_K^{(n)}(K) \ln p(X, Y, K) \right). \tag{17}$$

Next, we make use of the factorization of $p(X, Y, K)$ in order to simplify the log term inside the summation in (17). One natural factorization is $p(X, Y, K) = p(K, Y|X)p(X)$, where the first factor $p(K, Y|X)$ has already been discussed in (15). Proceeding to the prior term $p(X)$, we notice that it factorizes across targets: $p(X) = \prod_i p(X_i)$, since the individual target motions are assumed independent of other targets' evolution. Further, the Markovian property of individual target states sequences can also be used to factorize the prior term as, $p(X) = \prod_{t,i} p(X_{t,i}|X_{t-1,i})$. Using these simplifications in (17), we get

$$\begin{aligned} X^{(n+1)} = \arg\max_X \Big(& \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \\ & \times \exp\Big(\sum_{K,t,i} q_K^{(n)}(K) \ln p(Y_{t,j}, k_{t,i} = j|X_{t,i}) \Big) \Big). \end{aligned} \tag{18}$$

We next focus on simplifying the exponent term in (18), and observe that in summing over K, each $k_{t,i} \in \{0, 1, \ldots, M_t\}$. There can be several values of K, in which $k_{t,i} = j$, and we use the notation $K_{t,i,j}$ to denote these sets of hypotheses with assignment $k_{t,i} = j$. The implication of this analysis is that for a fixed value of j, $\ln p(Y_{t,j}, k_{t,i} = j|X_{t,i})$ appears in $K_{t,i,j}$ terms along with the weights $q_K^{(n)}$. Using this insight, the exponent in (18) can be rewritten as,

$$\sum_{t,i} \sum_{j=0}^{M_t} \left(\sum_{K_{t,i,j}} q_K^{(n)}(K) \right) \ln p(Y_{t,j}, k_{t,i} = j|X_{t,i}). \tag{19}$$

The term $\left(\sum_{K_{t,i,j}} q_K^{(n)}(K) \right)$ is in fact the marginal probability of assigning target i to measurement j, according to the distribution $q_K^{(n)}$. We use the notation,

$$w_{t,i,j}^{(n)} = \sum_{K_{t,i,j}} q_K^{(n)}(K) \tag{20}$$

to denote these marginal probabilities. Substituting the marginal probabilities in (18) and moving the summations out of the exponents, we get

$$X^{(n+1)} = \arg\max_{X} \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \times \exp\left(\sum_{j=0}^{M_t} w_{t,i,j}^{(n)} \ln p(Y_{t,j}, k_{t,i} = j|X_{t,i})\right). \quad (21)$$

Using the Gaussian process model for state transition density $p(X_{t,i}|X_{t-1,i})$ from (1), and also using (16) in (21), it further simplifies to

$$\begin{aligned} X^{(n+1)} &= \arg\max_{X} \prod_{i,t} \mathcal{N}\left(X_{t,i}; FX_{t-1,i}, Q\right) \\ &\quad \times \mathcal{N}\left(\tilde{Y}_{t,i}^{(n)}; HX_{t,i}, \tilde{R}_{t,i}^{(n)}\right), \end{aligned} \quad (22)$$

where, $\tilde{Y}_{t,i}^{(n)}$ and $\tilde{R}_{t,i}^{(n)}$ are the composite measurements and composite measurement covariances, given as,

$$\tilde{Y}_{t,i}^{(n)} = \frac{\sum_{j=1}^{M_t} w_{t,i,j}^{(n)} Y_{t,j}}{1 - w_{t,i,0}^{(n)}} \quad (23)$$

$$\tilde{R}_{t,i}^{(n)} = \frac{R}{1 - w_{t,i,0}^{(n)}}. \quad (24)$$

Details of this simplification are provided in Appendix A. Note that $\sum_{j=0}^{M_t} w_{t,i,j}^{(n)} = 1$. When the weight $w_{t,i,0}^{(n)}$ becomes 1 in (23) and (24), the likelihood term $\mathcal{N}\left(\tilde{Y}_{t,i}^{(n)}; HX_{t,i}, \tilde{R}_{t,i}^{(n)}\right) = 1$ in (22).

From (22), it can be inferred that the objective function of each target state $X_{t,i}$ is proportional to a Gaussian density. First factor in (22) is the prior term, while the second factor is likelihood of the composite measurement. A straightforward method to obtain these marginal posterior densities is by performing forward-backward smoothing, and the resulting mean yields the MAP estimates of state variables.

From (16), (20) and (22), it can be seen that the link between the E and M steps are through the marginal probabilities. The state estimate $X^{(n+1)}$ in (22) depends on the distribution $q_K^{(n)}$ in (16) through the marginal probabilities $w_{t,i,j}^{(n)}$ in (20). In the next section, we elaborate on the role of marginal probabilities in (20), and show how the LBP method in [37] can be employed to approximate them.

4 Marginal data association probabilities calculation using LBP

In this section, we discuss the challenges involved in computing the marginal data association probabilities $w_{t,i,j}^{(n)}$, that appear in (22) as part of the proposed algorithm. We also provide a brief overview of techniques to compute or approximate these probabilities. In this paper, we use loopy belief propagation (LBP) for computing these marginal probabilities. This LBP algorithm is inspired by the method employed in [37] for JPDA filtering. We extend it to compute marginal data association probabilities efficiently for batch problems.

To compute the marginal probabilities in (20), we make use of the expression for the data hypothesis probabilities in (16). The constraints that are implicitly in (16) due to the point target model assumptions pose the main challenge in computing the marginal probabilities in (20). At each time instant, there are constraints on the $k_{t,i}$ variables introducing dependency across targets. The constraint on $k_{t,i}$ at every time instant t ensures that no two targets share the same measurement at the same time. That is, $k_{t,i_1} \neq 0$ and $k_{t,i_2} \neq 0$ for $i_1 \neq i_2$ implies that $k_{t,i_1} \neq k_{t,i_2}$. However, no such constraints apply across time, that is, for targets i_1 and i_2, at different time instants, t_1 and t_2, the data association variables k_{t_1,i_1} and k_{t_2,i_2} are independent. Due to these constraints on the $k_{t,i}$ variables, $q_K(K)$ in (16) only factorizes across time, but not across targets within every time instant. Consequently, the marginalization of the data association variable can be performed at each time independently. Even so, the number of possible data associations at each time is very high, and it is computationally expensive to perform marginalization through an exhaustive listing of different possible values of K. Instead, there are less computationally intensive methods to obtain the marginal probabilities.

Techniques like k-best hypotheses based on Murty's algorithm [5, 24], Monte Carlo sampling method [23] and the tree-based approach [15] can be used to compute joint probabilities without explicit enumeration of all possible K. The method in [15] gives exact probability values whereas the methods in [5, 24] and [23] give approximations. The method we have employed to compute $w_{t,i,j}$ in (20) in our implementation is inspired by the graph based approach, loopy belief propagation presented in [37], which has been shown to provide good accuracy versus computation time tradeoff. In the next subsection, we present how the method proposed in [37] can be employed to the proposed algorithm. For the sake of brevity, we will drop the iteration indices.

4.1 Loopy belief propagation algorithm

The idea behind applying LBP is to use a factorization of $q_K(K)$ and apply the sum-product algorithm [18] to compute the marginal probabilities. In fact, in [37], a redundant representation for $q_K(K)$ is introduced and a factorization of this new representation is used to compute the marginals more accurately compared to what would have been possible with applying LBP on $q_K(K)$ directly. One intuition behind why this redundant representation works with LBP is that it makes the network for LBP bipartite which has slightly longer loops, as is shown in Figure 1; without the redundant variables, the network will be a complete graph (where all the nodes in the graph are connected to each other), which can possibly make the marginal approximation by LBP less accurate [37].

The LBP method proposed in [37] is for computing the marginal probabilities for JPDA filtering, whereas in this paper, the marginal probabilities are calculated for the batch problem where estimates from the smoothed posterior density. Consider the distribution of data association in (16), with the point target constraints expressed explicitly in the equation using factors denoted $\phi_{t,i,i'}$:

$$q_K(K) \propto \prod_{t,i} \psi_{t,i}(k_{t,i}) \prod_{j=i+1}^{N_T} \phi_{t,i,i'}(k_{t,i}, k_{t,i'}) \tag{25}$$

where

$$\psi_{t,i}(k_{t,i}) = \left(\frac{P_D \mathcal{N}(Y_{t,k_{t,i}}; HX_{t,i}, R)}{(1-P_D)\beta_c} \right)^{\mathbf{1}_{k_{t,i}}}, \tag{26}$$

$$\phi_{t,i,i'} = \begin{cases} 0 & k_{t,i} = k_{t,i'} \neq 0 \\ 1 & \text{otherwise} \end{cases}. \tag{27}$$

In our problem, the factors $\psi_{t,i}$ depend on the smoothed estimates of $X_{t,i}$, whereas in [37], they depend on the estimates from the prediction density. However, the constraints on the variable $k_{t,i}$

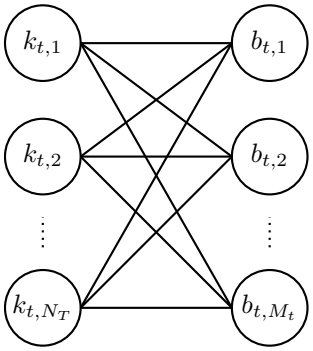


Figure 1: Belief propagation makes use of the dependencies across the (redundant) data association variables. As can be seen, the graph is bipartite which implies that the message flow is from the set of nodes on the left to the right and vice versa, when performing LBP.

are the same in both the problems. So, the LBP method in [37] is applicable here wherein we use the smoothed estimates of $X_{t,i}$ instead of the predicted estimates.

Similar to [37], another set of variables $B = \{b_{t,j}, \forall t, j\}$ is introduced such that we have a redundant representation $q_{K,B}(K, B)$ instead of $q_K(K)$. The $b_{t,j}$ variables are the measurement-oriented data association variables, i.e., $b_{t,j}$ stands for assigning the measurement $Y_{t,j}$ to the target $b_{t,j}$ at time t. Using these variables, $q_{K,B}(K, B)$ can be comprehensively expressed, including the constraints as follows:

$$q_{K,B}(K, B) \propto \prod_{t,i} \psi_{t,i}(k_{t,i}) \prod_{j=1}^{Mt} \psi_{t,i,j}(k_{t,i}, b_{t,j}), \tag{28}$$

where

$$\psi_{t,i,j}(k_{t,i}, b_{t,j}) = \begin{cases} 0, & \substack{k_{t,i}=j, b_{t,j}\neq i \\ k_{t,i}\neq j, b_{t,j}=i} \\ 1, & \text{otherwise} \end{cases}. \tag{29}$$

Interpretation of the function $\psi_{t,i,j}(k_{t,i}, b_{t,j})$ is that it takes the value '1' when the target-oriented variable $k_{t,i}$ and the measurement-oriented variable $b_{t,j}$ agree regarding the association and the value '0' when there is a disagreement. So, these $\psi_{t,i,j}$ functions capture the constraints that a target can be assigned to at most one measurement and a measurement can be assigned to at most one target. The important advantage of introducing these new functions in the representation of $q_K(K)$ is that the dependency among the variables $k_{t,i}$ is only through the variables $b_{t,j}$. This dependency is best illustrated in the factor graph for (28) in Figure 1. The graph as can be seen is bipartite. This is advantageous because the possible message flow in LBP will be from the nodes on the left side of the graph to the right side of the graph and vice versa.

Once the factorization with $\psi_{t,i}$ and $\psi_{t,i,j}$ functions is established, one can follow the procedures discussed in Sections III-A and III-B of [37] to arrive at the expressions for the messages. The messages passed during the iteration l from $k_{t,i}$ variables to $b_{t,j}$ variables are

$$\eta_{t,i\to j}^{(l)} = \frac{\psi_{t,i}(k_{t,i} = j)}{1 + \sum\limits_{j'\neq j, j'>0} \psi_{t,i}(k_{t,i} = j')\nu_{t,j'\to i}^{(l-1)}} \tag{30}$$

and the messages passed from $b_{t,j}$ variables to $k_{t,i}$ variables are

$$\nu_{t,j\to i}^{(l)} = \frac{1}{1+\sum\limits_{i'\neq i, i'>0} \eta_{t,i'\to j}^{(l)}}. \tag{31}$$

The messages are passed in both directions until convergence. Upon convergence, it follows from [37] that the marginal probabilities are calculated according to

$$w_{t,i,j} = \frac{\psi_{t,i}(k_{t,i}=j)\nu_{t,j\to i}^{(l)}}{\sum\limits_{j'} \psi_{t,i}(k_{t,i}=j')\nu_{t,j'\to i}^{(l)}}. \tag{32}$$

A step-by-step algorithmic description for every time instant t is given in Algorithm 1.

Algorithm 1 Loopy belief propagation algorithm for calculating marginal data association probabilities

Input: $\psi_{t,i}(j)$, $i = 1, \ldots, N_T$, $j = 1, \ldots, M_t$
Set $l = 0$.
Initialize $\nu_{t,j\to i}^{(l)} = 1$.
while Not converged **do**
  Set $l = l + 1$.
  for $i = 1, \ldots, N_T$, $j = 1, \ldots, M_t$ **do**
    Compute $\eta_{t,i\to j}^{(l)}$ according to (30).
  end for
  for $i = 1, \ldots, N_T$, $j = 1, \ldots, M_t$ **do**
    Compute $\nu_{t,j\to i}^{(l)}$ according to (31).
  end for
end while
for $i = 1, \ldots, N_T$, $j = 0, \ldots, M_t$ **do**
  Compute $w_{t,i,j}$ according to (32).
end for
Output: $w_{t,i,j}$, $i = 1, \ldots, N_T$, $j = 0, \ldots, M_t$

5 Algorithm

In this section, we present implementation details of the proposed algorithm for the Gaussian model assumptions. We discuss how the equations and methods derived in Section 3, and Section 4 can be interpreted, and implemented using existing algorithms in the literature. We present a step-by-step algorithm that describes the proposed solution.

The E and M steps in the proposed algorithm involve obtaining the marginal data association probabilities and MAP estimates of the state variable, as described in Section 3. To initialize the algorithm, we assume that the estimates $X_{t,i}^{(0)}, \forall t, i$ are known. In our implementation, we use the true target states at time zero, i.e., for $X_{0,i}^{(0)}$ and run Kalman filters using the composite measurements, whose weights are obtained using the LBP, as described in Section 4 and obtain $X_{t,i}^{(0)}$ for all t and i.

In the E-step, we compute $\psi_{t,i}$ according to (26). Given these values, the marginal probabilities $w_{t,i,j}^{(n)}$ are computed using the LBP method in Algorithm 1. With these marginal probabilities, the composite measurements and covariances in (23) and (24), respectively, are computed.

The M-step in (22) involves forward-backward smoothing for each target i with the likelihood $\mathcal{N}\left(\tilde{Y}_{t,i}; HX_{t,i}, \tilde{R}_{t,i}\right)$ of these composite measurements and covariances. In our implementation, forward-backward smoothing is carried out using RTS smoother (described in Appendix B) which provide the desired MAP estimates $X_{t,i}^{(n)}$ in (22). Note that the complexity of running the RTS smoothers, thus, scales linearly with the number of targets. The new estimates $X_{t,i}^{(n)}$ from the RTS smoother are passed on to the next iteration of E-step, and the algorithm is run until convergence.

One can use common ways of checking for convergence, for instance, to check for the change in estimates after each iteration. In our implementation, when the change is sufficiently small, we terminate the algorithm. The entire algorithm is summarized in Algorithm 2 .

Algorithm 2 Proposed multi-target tracking algorithm based on EM

Input: Measurements Y

Initialize $X_{t,i}^{(n)}$, $t = 1, \ldots, T$, $i = 1, \ldots, N_T$ for $n = 0$.
while not converged **do**
  E step
  for $t = 1, \ldots, T$ **do**
    Compute $\psi_{t,i}^{(n)}(k_{t,i})$ $\forall\, i$ and $\forall\, k_{t,i} = 1, \ldots, M_t$ according to (26).
    Marginal probabilities: Compute $w_{t,i,j}^{(n)}$ according to (32) using loopy belief propagation, described in Algorithm 1.
  end for
  M step
  Composite measurements: With inputs $w_{t,i,j}^{(n)}$, compute $\tilde{Y}_{t,i}$, $\tilde{R}_{t,i}\, \forall\, t, i$, according to (23) and (24).
  for $i = 1, \ldots, N_T$ **do**
    RTS Smoothing: With inputs $X_{0,i}^{(n)}, P_{0,i}^{(n)}, \tilde{Y}_{t,i}, \tilde{R}_{t,i}$ compute the means $X_{t,i}^{(n)}\, \forall\, t$ using RTS smoother described in Algorithm 3.
  end for
  Set $X_{t,i}^{(n+1)} = X_{t,i}^{(n)}, \forall t, i$.
  Set $n = n + 1$.
end while

Output: $X_{t,i}^{(n)}, w_{t,i,j}^{(n)}\, \forall\, t, i, j$

6 Comparison with PMHT and JPDA

In this section, we compare the proposed method with PMHT and JPDA to highlight some of the similarities and differences using example illustrations. The comparison with PMHT is presented based on marginal probabilities, where we show that PMHT can coalesce trajectories as it does not enforce the point target constraints. Similar coalescence property is also exhibited by JPDA [9], but in this case it is due to moment matching.

6.1 Comparison with PMHT

As pointed in Section 2.2, the proposed algorithm is similar to PMHT, as both algorithms are based on EM and treat the data association variables as the hidden variables. The similarity becomes even more pronounced in the algorithmic details, as both perform smoothing with composite measurements of the form in (23) and covariances as in (24). Further, local hypothesis probability

calculation is identical as in (26). However, the two algorithms differ in the computation of marginal probabilities for composite measurements from the local hypothesis probabilities.

PMHT assumes that a target can be assigned to multiple measurements, implying that there are no constraints on the data association variables $k_{t,i}$, and distribution over the data association variables, $q_K(K)$, in (25) factorizes across time and targets. Consequently, the weights for composite measurements are computed directly from local hypothesis probabilities, $\psi_{t,i}(k_{t,i})$, after normalizing them as probabilities. On the other hand, in the proposed EM algorithm, there are constraints on $k_{t,i}$ across targets at each time as was discussed in Section 4. As a result, calculation of the weights is performed conforming to the constraints, which in this work is implemented using the LBP method presented in Section 4.1. In summary, the proposed EM algorithm with an unconstrained marginal probability computation (but with normalized $\psi_{t,i}$) would be converted into the PMHT algorithm.

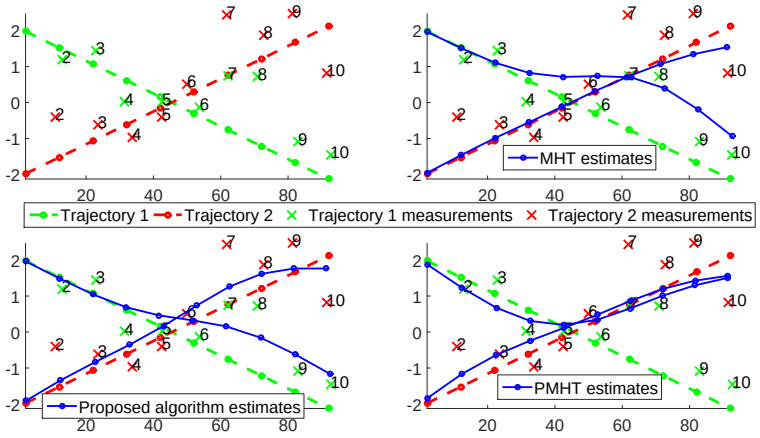


Figure 2: The figure shows slowly moving, crossing target trajectories, along with their corresponding measurements. Numbers adjacent to the measurements indicate time index of the corresponding measurements. Also shown are the trajectory estimates returned by N-scan pruning based TOMHT with a scan depth of 4, the proposed algorithm and PMHT.

One possible implication of relaxing the point target model constraints in PMHT, when computing marginal probabilities, is track coalescence. We illustrate this using the crossing targets example, depicted in Figure 2, and compare it to the proposed EM algorithm and an N-scan pruning based TOMHT algorithm with a scan depth. The figure shows two crossing trajectories generated with zero process noise; the measurements shown are generated by setting $P_D = 1$ and $\beta_c = 0$. It also shows the trajectory estimates (with process noise standard deviation of $0.2165m$) returned by PMHT, the proposed EM algorithm and TOMHT. From the figure, it is evident that PMHT algorithm coalesces both trajectories after the crossing point. This behaviour can be best explained by observing the marginal probabilities, $\omega_{t,i,j}$, returned by PMHT, when the targets cross. For instance, the marginal probabilities at $t = 6$ are shown in Table 1. As it has been emphasized before, PMHT can allow targets to be associated to the same measurement; in this case, both targets get high weights for measurement 2, and get updated with the same measurement, resulting in similar trajectories. On the other hand, weights returned by the proposed algorithm are shown in Table 2, wherein, we see that it has ruled out the possibility of assigning the same measurement to both targets and thus, target 1 has high weight for measurement 1 and target 2 for measurement 2. Thus, the targets are kept separated when there are well-spaced measurements as shown in Figure 2. It can also be observed that the estimates from the proposed algorithm are very close to the estimates from TOMHT algorithm, which we use as the benchmark.

$\omega_{t,i,j}$	Measurement 1	Measurement 2
Target 1	0.06	0.93
Target 2	0.08	0.92

Table 1: The table shows values of marginal probabilities $\omega_{t,i,j} \propto \psi_{t,i}(j)$ returned by PMHT at $t = 6$.

$\omega_{t,i,j}$	Measurement 1	Measurement 2
Target 1	0.987	0.013
Target 2	0.077	0.923

Table 2: The table shows values of marginal probabilities $\omega_{t,i,j}$ returned by EM algorithm at $t = 6$.

6.2 Comparison with JPDA

The similarities between JPDA [5, 12] and the proposed EM algorithm arise from the fact that both algorithms use composite measurements to update target states at each time instant, which requires computation of the marginal probabilities. One can therefore use similar strategies to compute the marginal probabilities for both algorithms, see Section 4) for a discussion on these strategies. However, there is a fundamental difference in the objective of the two algorithms, as pointed out in Section 2.2. JPDA is primarily a filtering algorithm that approximates the posterior density $p(X_t|Y_{1:t})$ by moment matching, whereas the proposed EM algorithm is a batch solution that provides MAP estimates of the states.A shortcoming of the recursive moment matching in JPDA is that it can lead to coalescence when there are closely moving targets.

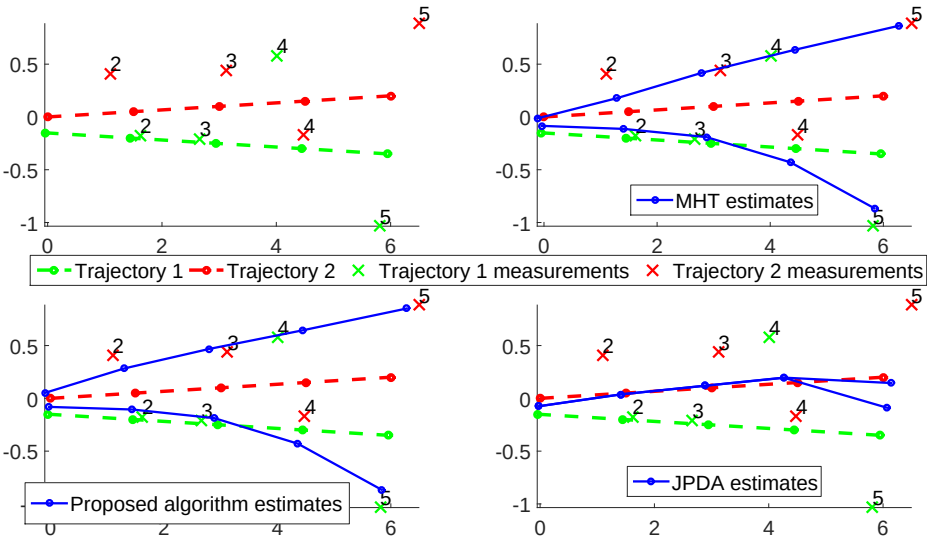


Figure 3: The figure shows two closely moving targets with the corresponding measurements. Also shown are the trajectories estimated by MHT, the proposed EM algorithm and JPDA.

We illustrate the coalescence effect in JPDA using an example shown in Figure 3. The figure shows two targets that are moving slowly in close proximity of each other, along with the measurements that are generated with $P_D = 1$ and $\beta_c = 0$. For this example, JPDA, TOMHT and the proposed

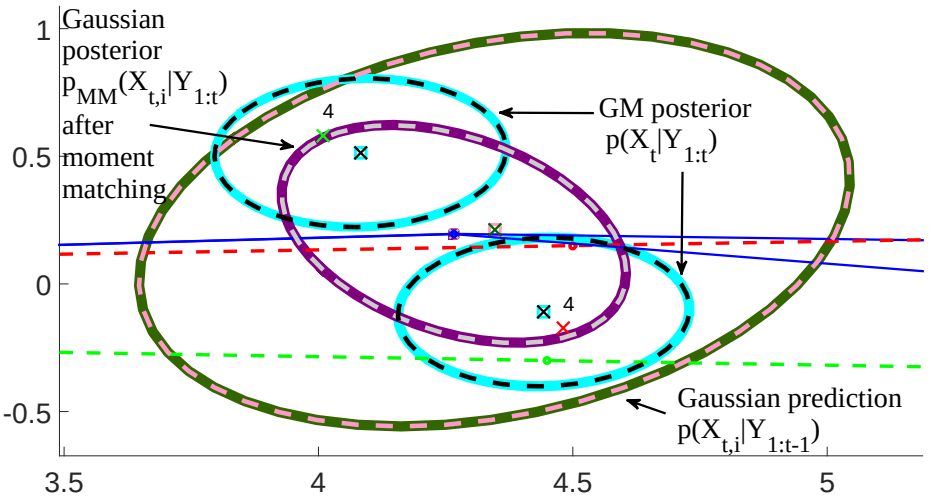


Figure 4: The figure shows the illustration of densities showing the coalescence effect in JPDA due to moment matching at time $t = 4$ for the example in Figure 3. The pink and dark green ellipses represent the Gaussian prediction density for the two targets. The cyan and black pairs of ellipses represent the two Gaussian components in the joint posterior density of the two targets. The purple and the grey ellipses show the posterior density of the two target states after moment matching.

algorithm have the same initialization for $X_{0,i}$; the values of $X_{0,i}$ are very close for the two targets $i = 1, 2$. Figure 3 shows the trajectories returned by the three algorithms. It can be observed that MHT and the proposed algorithm return almost identical results whereas JPDA has coalesced the two trajectories. The coalescence in JPDA is due to moment matching, which can be explained by looking closer at the densities, for instance, at time $t = 4$ (Figure 4). Due to close initiations, the prediction densities $p(X_{t,i}|Y_{1:t-1})$ at $t = 4$ for both the targets are almost identical for both targets. The similarity between the densities is illustrated by the overlapping pink and dark green ellipses in Figure 4. Given this prior, there are two possible hypotheses – target 1 is assigned to measurement 1, whereas target 2 assigned to measurement 2 and vice versa – leading to a Gaussian mixture posterior density $p(X_t|Y_{1:t})$ with two components, corresponding to the two hypotheses. However, since the priors are very similar, the two components corresponding to the two hypotheses are almost identical, shown by the overlapping cyan and black ellipses in the figure. Given this posterior density, any MAP estimation algorithm would pick the mean of one of the components, i.e., either the cyan or the black. However, JPDA does a moment matching of the two components and arrives at a Gaussian posterior density $p_{MM}(X_{t,i}|Y_{1:t})$ for the two target states, which are again almost identical, shown by the purple and grey ellipses in the figure. From this illustration, it is clear that the JPDA algorithm tends to coalesce close trajectories even if the measurements are far apart, due to moment matching of the GM posterior density. A MAP estimation algorithm like the proposed algorithm or TOMHT, on the other hand, outputs the mode of the posterior, thus avoiding coalescence as shown in Figure 3.

7 Simulations and results

In this section, we evaluate the proposed algorithm for different scenarios and compare its performance with PMHT, JPDA and N-scan pruning based TOMHT with scan depth 3. In Section 7.1,

Scenario			Position estimates				Velocity estimates			
			TOMHT	Prop.Alg.	PMHT	JPDA	TOMHT	Prop.Alg.	PMHT	JPDA
$P_D = 0.9$,	$\beta_c =$	0.5	1.19	0.79	2.76	2.57	0.43	0.34	0.95	0.75
$N_T = 20$		2.5	1.35	1.25	2.71	2.75	0.48	0.46	0.88	0.78
		4.5	1.47	1.58	2.55	2.97	0.50	0.56	0.81	0.82
$P_D = 0.9$,	$N_T =$	10	0.94	1.53	3.27	3.86	0.44	0.59	0.80	0.89
$\beta_c = 1.5$		20	1.23	1.30	2.35	2.61	0.43	0.47	0.78	0.76
		30	1.32	0.98	1.77	1.88	0.42	0.34	0.61	0.68
$N_T = 20$,	$P_D =$	0.8	1.46	1.51	2.34	2.62	0.51	0.52	0.78	0.79
$\beta_c = 1.5$		0.9	1.08	1.15	2.59	2.79	0.39	0.43	0.76	0.81
		0.99	0.88	0.60	2.56	2.97	0.35	0.31	0.73	0.94

Table 3: The table shows RMSE per time scan for position estimates (in m), and velocity estimates (in m/s) based on the MOSPA assignment. The values for each of the scenarios are averaged over 1000 Monte Carlo trials. The values of β_c shown in the table are scaled by 10^4, i.e., in the range of 0.5×10^{-4} to 4.5×10^{-4}.

we first describe the simulation set-up in terms of model parameter values, comparison metrics and methods of comparison. In Section 7.2, we briefly describe two techniques used for handling the sensitivity to initialization of the proposed algorithm. In Section 7.3, we evaluate the impact of initialization techniques on the proposed algorithm. Computation of marginal data association probabilities using LBP is compared with brute force approach to show the difference in performance of the proposed algorithm. Finally, a comprehensive comparison based on MOSPA and computation time of the four aforementioned algorithms is presented.

7.1 Simulation scenarios

To evaluate the performance of the proposed EM algorithm, TOMHT, PMHT and JPDA, we have used a four-dimensional constant-velocity model with additive discrete-time noise, with parameters,

$$X_{t,i} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, F = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$Q = \sigma_v^2 \begin{bmatrix} T_s^4/4 & 0 & T_s^3/2 & 0 \\ 0 & T_s^4/4 & 0 & T_s^3/2 \\ T_s^3/2 & 0 & T_s^2 & 0 \\ 0 & T_s^3/2 & 0 & T_s^2 \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ and } R = \sigma_w^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

In the above equations, T_s is the sampling time. The values of the parameters are $T_s = 0.5$ s, $\sigma_v = 8m/s^2$ and $\sigma_w = \sqrt{5}m$ for all scenarios and the batch length T is 20. The initial velocity for the constant velocity model was set at 20 m/s on both the x and y directions. The x- and y-coordinates of the initial points $X_{0,i}$, $i = 1, \ldots, N_T$ for the targets were chosen uniformly from a region of width 80m. To simulate different scenarios, the following combinations of parameters have been used:

- $P_D = [0.8, 0.9, 0.99]$, $\beta_c = 1.5 \times 10^{-4}$ and $N_T = 20$
- $\beta_c = [0.5, 2.5, 4.5] \times 10^{-4}$, $P_D = 0.9$ and $N_T = 20$
- $N_T = [10, 20, 30]$ $P_D = 0.9$ and $\beta_c = 1.5 \times 10^{-4}$.

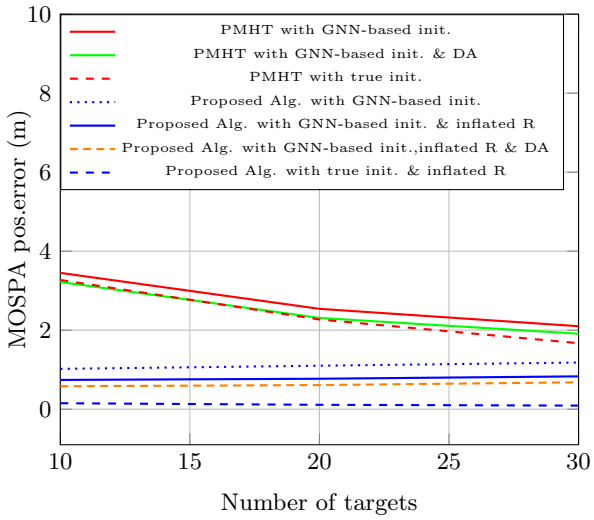


Figure 5: Comparison of MOSPA based position error for the proposed algorithm and PMHT, with GNN based initialization and true state initialization. The impact of deterministic annealing and inflated measurement noise on the MOSPA performance of the proposed algorithm is also shown.

We have used Euclidean distance based MOSPA [14, 31] and computation time as the metrics to evaluate and compare performances. MOSPA calculation requires the minimum mean squared error between the set of true tracks and estimated tracks, which turns out to be expensive to compute for high number of targets. We employ the modified auction algorithm [27], by posing the MOSPA calculation as a 2-D assignment problem that attempts to minimize the overall assignment cost (MOSPA). To compare the computation time, all four algorithms were run on a desktop computer with quad-core processor and 8 gigabyte RAM, running Debian operating system.

7.2 Handling sensitivity to initialization

The proposed algorithm and PMHT are based on EM, and it is well known that EM algorithm is sensitive to initialization [33, 34] and can converge to a local optimum. To deal with this issue, we use two common strategies in our implementation: 1) Deterministic annealing [33] 2) Homotopy methods [25]. In this section, we describe these methods briefly and show empirically that these methods improve the performance compared to naive initializations.

In the deterministic annealing (DA) [33] strategy, the idea is to compute the E-step with $(q_K^{(n)})^\beta$ instead of $q_K^{(n)}$, where $0 < \beta \leq 1$. The value of β is increased successively, from a small value to begin with, reaching 1 when EM iterations stop. In our algorithm 2, introducing the deterministic annealing only affects the Step 5; before passing ψs to the next step, we raise these values by the exponent β in the current iteration.

Inspired from the homotopy methods [25], we use a strategy to run several instances of the algorithm with varying measurement noise covariance R, which we refer to as the inflated R strategy. The value of R is changed only within the algorithm, and not in the generation of the measurements. To begin with, the measurement noise covariance is set to a higher value compared to the true value and the entire EM algorithm is run with this inflated R. After the EM algorithm converges with the current inflated R, the estimates from this run of the algorithm are used as the initialization for the next run of the algorithm. This procedure is repeated until R is reduced to its true value.

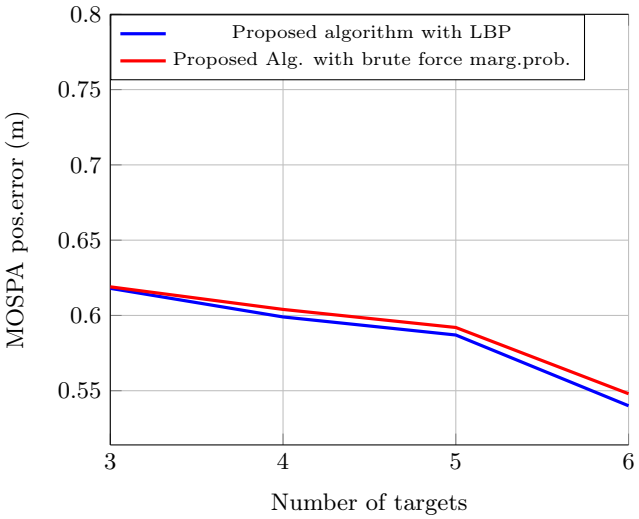


Figure 6: Comparison of MOSPA based position error for the proposed algorithm with brute force compuation of marginal data association probabilities and the approximation using LBP.

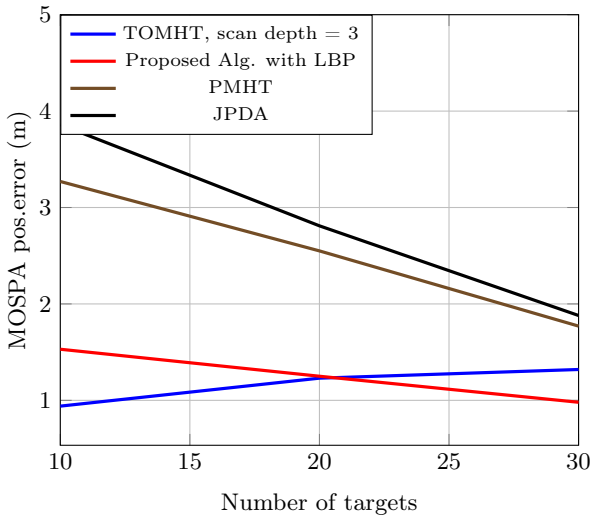


Figure 7: Comparison of MOSPA based position error for the different algorithms considered, for varying number of targets scenario corresponding to the first row in Table 3.

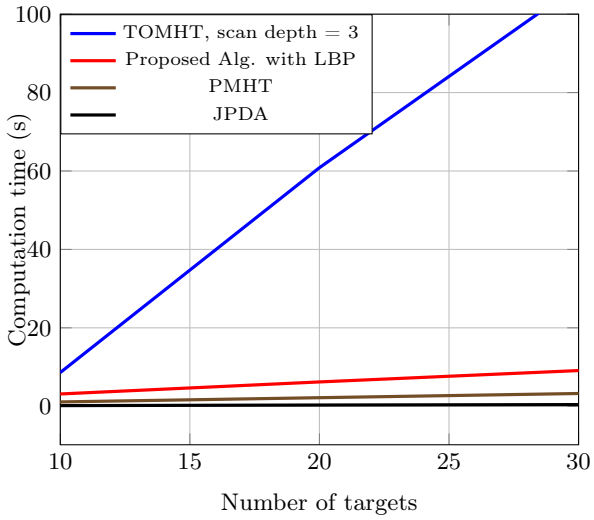


Figure 8: Computation time comparison for the different algorithms considered, for varying number of targets scenario corresponding to the first row in Table 3.

7.3 Results

In this subsection, we present the simulation results highlighting different aspects of the proposed algorithm. First, we present the impact of using homotopy methods and DA on the performance of the proposed algorithm. Second, we compare the quality of the approximation of the marginal probabilities using LBP in the proposed algorithm with the brute force approach. Finally, we compare the MOSPA and computation time performances of the proposed algorithm with TOMHT, PMHT and JPDA for varying set of parameters. All simulation results presented are averaged over 1000 Monte Carlo runs.

7.3.1 Impact of homotopy methods and DA

The gains achieved by using homotopy methods and DA for the proposed algorithm are large compared to those achieved by varying PMHT initializations. To highlight these differences, performance of both the algorithms implemented with these methods is compared against two other naive (without homotopy and deterministic annealing) strategies: the naive implementation with the true target states as initialization, and the naive implementation with a GNN-based initializations. The observations are presented in Figure 5 for $N_T = [10, 20, 30]$, $\beta_c = 1.5 \times 10^{-4}$ and $\sigma_v = 6m/s^2$. From Figure 5, it is clearly seen that the proposed algorithm benefits from DA and inflated R strategy. It shows further improved performance when initialized with true target states. On the contrary, we observe that PMHT shows no such improvement even when initialized with true target states. In summary, the proposed algorithm in all its variations considered shows improved performance when compared to PMHT, and we also expect it to improve further with better initializations.

7.3.2 Comparison of brute force and LBP computation of marginal probabilities

In Section 4, we presented LBP as an efficient method to approximate the brute force computation of marginal data association probabilities. We substantiate this approximation by comparing the difference in performance when the proposed algorithm uses LBP and brute force method for the same scenarios. We have compared the difference in the position RMSE for the two instances of

the proposed algorithm, one with LBP and the other with brute force for $N_T = [3, 4, 5, 6]$, limited by the massive computation time of brute force approach. The results are presented in Figure 6, where we can observe that there is no significant difference in performance of the proposed algorithm for both methods of computing marginal data association probabitlies. Based on this observation, we conclude that the approximation of marginal data association probabilities using LBP does not degrade the performance of the proposed algorithm, despite providing large (from exponential to quadratic) computational advantage.

7.3.3 Performance evaluation of the proposed algorithm

In Table 3, we present a comprehensive comparison of the MOSPA results for different scenarios for the different algorithms. The table shows the mean values of root mean squared error in position and velocity components of the estimated states. The table captures the performance trend of each of the algorithms in the scenarios considered. For instance, the first row of numeric entries correspond to the scenario when $P_D = 0.9, N_T = 20, \beta_c = 0.5 \times 10^{-4}$, and we notice that the mean position and velocity errors for the proposed algorithm are lower than the other algorithms. Although not reported in the table, we have observed a similar trend of lower standard deviation of the position and velocity errors for the proposed algorithm, by which we infer that the proposed algorithm is more robust than JPDA and PMHT for the scenarios considered. MOSPA based position error for the varying number of targets scenario is presented in Figure 7. From the figure it can be seen that the proposed algorithm performs better than PMHT and JPDA for all scenarios. For dense scenarios, the proposed algorithm even outperforms the TOMHT considered.

The computation time comparison is performed based on the run time of the algorithm for varying number of targets, shown in Figure 8. The results presented capture the trend in computation time of the proposed algorithm in comparison to the existing algorithms. It should be noted that the implementation of all the four algorithms can be optimized for run time; for instance, one can parallelize the implementations. In that case, we expect the graph shown in Figure 8 to scale down by an appropriate factor. Nevertheless, it can be seen that the proposed algorithm's run time is less than our implementation of TOMHT. The proposed algorithm delivers improved MOSPA performance in comparison to PMHT. By employing LBP to compute marginal probabilities, the computation time only increases marginally when compared to JPDA (that also uses LBP) and PMHT. In summary, the proposed algorithm yields performance close to TOMHT, but with a significant computational advantage.

8 Conclusions and future work

In conclusion, we have presented a multi-target tracking algorithm based on expectation maximization, and marginal data association probabilities. We have shown that our solution for tracking point targets uses the correct point target model assumptions and does not violate these assumptions in deriving the solution. The solution has been presented for the case of known number of targets and for batch problems, and has been compared comprehensively with JPDA, PMHT and TOMHT, highlighting the prominent differences between them. The performance of the proposed algorithm has been evaluated for varying number of targets, clutter intensities and probability of detection, and we have shown that for the scenarios considered, our algorithm performs considerably better than JPDA and PMHT in MOSPA sense, and is computationally more inexpensive than a basic TOMHT.

We see substantial potential in the proposed algorithm, and envisage many possible enhancements to it. While we have discussed the proposed algorithm as a batch solution, one evident limitation with this approach, as with PMHT, is it not being an online solution, which might be a requirement

in many tracking applications. Another possible extension is to accommodate target birth/death into the solution framework, to complete the tracking solution. One of the commonly used strategies to initialize new targets is measurement-driven track initialization. A more coherent approach to track handling is to model target existence uncertainties as a random variable and estimating it. The algorithm has been observed to be sensitive to initialization. So, further improvement to initialization techniques would be desirable as well.

Appendices

A Derivation of the EM algorithm for the Gaussian case

In this section, the derivation of the EM algorithm for the assumptions made in Section 2 will be completed. That is, we will sketch the derivations starting from (21) to (22) in Section 3.

We begin from equation (21),

$$\begin{aligned} X^{(n+1)} &= \arg\max_X \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \\ &\quad \times \exp\left(\sum_{j=0}^{M_t} w_{t,i,j}^{(n)} \ln p(Y_{t,j}, k_{t,i}=j|X_{t,i})\right). \end{aligned} \tag{33}$$

Moving around the ln and exp, the above equation immediately simplifies to

$$\begin{aligned} X^{(n+1)} &= \arg\max_X \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \\ &\quad \times \left(\prod_{j=0}^{M_t} p(Y_{t,j}, k_{t,i}=j|X_{t,i})^{w_{t,i,j}^{(n)}}\right). \end{aligned} \tag{34}$$

Substituting the expressions for $p(Y_{t,j}, k_{t,i}=j|X_{t,i})$ and considering only the terms that depend on X, we get

$$\begin{aligned} X^{(n+1)} &= \arg\max_X \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \\ &\quad \times \left(\prod_{j=1}^{M_t} \mathcal{N}(Y_{t,j}; HX_{t,i}, R)^{w_{t,i,j}^{(n)}}\right). \end{aligned} \tag{35}$$

Note that $\mathcal{N}(Y_{t,j}; HX_{t,i}, R)^{w_{t,i,j}^{(n)}} \propto \mathcal{N}(Y_{t,j}; HX_{t,i}, \frac{R}{w_{t,i,j}^{(n)}})$. Along with this relation, the following result for the product of Gaussians can be used in (35),

$$\prod_{i=1}^{N} \mathcal{N}(x; \mu_i, P_i) \propto \mathcal{N}(x; \mu, P) \tag{36}$$

where $P = \left(\sum_{i=1}^{N} P_i^{-1}\right)^{-1}$ and $\mu = P\left(\sum_{i=1}^{N} P_i^{-1}\mu_i\right)$. After simplifications, we get

$$\begin{aligned} X^{(n+1)} &= \arg\max_X \prod_{i,t} \mathcal{N}(X_{t,i}; FX_{t-1,i}, Q) \\ &\quad \times \mathcal{N}\left(\tilde{Y}_{t,i}^{(n)}; HX_{t,i}, \tilde{R}_{t,i}^{(n)}\right). \end{aligned} \tag{37}$$

Equation (37) is same as the equation for the joint density of a single target, where instead of a simple measurement update, the update is performed with the composite measurement $\tilde{Y}_{t,i}^{(n)}$, given in (23), and with uncertainty $\tilde{R}_{t,i}^{(n)}$, given in (24).

B Smoothing with the RTS smoother

Rauch-Tung-Striebel (RTS) smoother is the ubiquitously used forward-backward smoothing solution. For linear Gaussian models, the forward filtering is performed using a Kalman filter, with the moments calculated as,

$$\mu_{t+1|t} = F\mu_t \tag{38}$$
$$P_{t+1|t} = FP_tF^T + Q \tag{39}$$
$$S_{t+1} = HP_{t+1|t}H^T + R \tag{40}$$
$$K_{t+1} = P_{t+1|t}H^T S_{t+1}^{-1} \tag{41}$$
$$\mu_{t+1} = \mu_{t+1|t} + K_t(y_{t+1} - H\mu_{t+1|t}) \tag{42}$$
$$P_{t+1} = P_{t+1|t} - K_{t+1}S_{t+1}^{-1}K_{t+1}^T \tag{43}$$

Backward smoothing is performed using the moments computed from the forward filter, with the following smoothing equations.

$$G_t = P_tF^TP_{t+1|t} \tag{44}$$
$$\mu_t^s = \mu_t + G_t(\mu_{t+1}^s - \mu_{t+1|t}) \tag{45}$$
$$P_t^s = P_t - G_t[P_{t+1|t} - P_{k+1}^s]G_t^T \tag{46}$$

Algorithm 3 RTS smoothing

Input: Initial estimates μ_0, P_0, and measurement data $y_t \; \forall \, t = 1, \ldots, T$
  for all $t = 1, \ldots, T$ **do**
    Compute and store the moments $\mu_{t|t-1}$, $P_{t|t-1}$, μ_t, P_t according to (38), (39), (42) and (43), respectively
  end for
  for all $t = T-1, \ldots, 1$ **do**
    Compute the smoothed moments μ_t^s, P_t^s according to (45) and (46), respectively
  end for

References

[1] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 618–626, 1978.

[2] ——, "Multitarget-multisensor tracking: advanced applications," *Norwood, MA, Artech House, 1990, 391 p.*, vol. 1, 1990.

[3] Y. Bar-Shalom and X.-R. Li, "Multitarget-multisensor tracking: Principles and techniques," *Storrs, CT: University of Connecticut, 1995.*, 1995.

[4] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London, 2003.

[5] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, Inc., 1999.

[6] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, 2004.

[7] E. A. Bloem and H. A. Blom, "Joint probabilistic data association methods avoiding track coalescence," in *IEEE Conference on Decision and Control*, vol. 3. IEEE, 1995, pp. 2752–2757.

[8] K.-C. Chang and Y. Bar-Shalom, "Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers," *IEEE Transactions on Automatic Control*, vol. 29, no. 7, pp. 585–594, 1984.

[9] S. Coraluppi, C. Carthel, P. Willett, M. Dingboe, O. O'Neill, and T. Luginbuhl, "The track repulsion effect in automatic tracking," in *International Conference on Information Fusion*. IEEE, 2009, pp. 2225–2230.

[10] D. F. Crouse, M. Guerriero, and P. Willett, "A critical look at the PMHT," *Journal of Advances in Information Fusion*, vol. 4, no. 2, pp. 93–116, 2009.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[12] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[13] A. F. Garcia-Fernandez, M. R. Morelande, and J. Grajal, "Bayesian sequential track formation," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6366–6379, 2014.

[14] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in *International Conference on Information Fusion*. IEEE, 2010, pp. 1–8.

[15] P. Horridge and S. Maskell, "Real-time tracking of hundreds of targets with efficient exact JPDAF implementation," in *International Conference on Information Fusion*. IEEE, 2006, pp. 1–8.

[16] J. Kokkala and S. Särkkä, "Combining particle MCMC with Rao-Blackwellized Monte Carlo data association for parameter estimation in multiple target tracking," *arXiv preprint arXiv:1409.8502*, 2014.

[17] D. Koller, J. Weber, and J. Malik, *Robust multiple car tracking with occlusion reasoning*. Springer, 1994.

[18] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[19] T. Kurien, M. Liggins *et al.*, "Report-to-target assignment in multisensor multitarget tracking," in *IEEE Conference on Decision and Control*. IEEE, 1988, pp. 2484–2488.

[20] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *International Workshop on Intelligent Robots and Systems*. IEEE, 1991, pp. 1442–1447.

[21] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[22] T. Minka, "Divergence measures and message passing," Technical report, Microsoft Research, Tech. Rep., 2005.

[23] M. R. Morelande, "Joint data association using importance sampling," in *International Conference on Information Fusion*. IEEE, 2009, pp. 292–299.

[24] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Research*, vol. 16, no. 3, pp. 682–687, 1968.

[25] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[26] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *IEEE Conference on Decision and Control*, vol. 1. IEEE, 2004, pp. 735–742.

[27] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Transactions on Automatic Control*, vol. 37, no. 2, pp. 198–213, 1992.

[28] G. Pulford, "Taxonomy of multiple target tracking methods," in *IEE Proceedings - Radar, Sonar and Navigation*, vol. 152, no. 5. IET, 2005, pp. 291–304.

[29] H. E. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[30] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[31] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[32] R. L. Streit and T. E. Luginbuhl, "Probabilistic multi-hypothesis tracking," DTIC Document, Tech. Rep., 1995.

[33] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Neural Networks*, vol. 11, no. 2, pp. 271–282, 1998.

[34] B. Wang, D. Titterington *et al.*, "Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model," *Bayesian Analysis*, vol. 1, no. 3, pp. 625–650, 2006.

[35] P. Willett, Y. Ruan, and R. Streit, "PMHT: Problems and some solutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 738–754, 2002.

[36] J. L. Williams and R. A. Lau, "Data association by loopy belief propagation," in *International Conference on Information Fusion*. IEEE, 2010, pp. 1–8.

[37] J. Williams and R. Lau, "Approximate evaluation of marginal association probabilities with belief propagation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2942–2959, 2014.

Paper B

Expectation maximization with hidden state variables for MTT

R.Selvan, A.S.Rahmathullah and L.Svensson

To be submitted

Expectation maximization with hidden state variables for multi-target tracking

Raghavendra Selvan, Abu Sajana Rahmathullah, Lennart Svensson

Abstract

In this paper, we present an algorithm for tracking multiple point targets based on expectation maximization (EM). Unlike the probabilistic multiple hypotheses tracking (PMHT) algorithm which is also based on EM, in this paper we treat target states as hidden variable; in PMHT data association variable is treated as hidden variable. We show that the proposed algorithm can be implemented as iterations of Rauch-Tung-Striebel (RTS) smoothing for estimating target states and 2-D auction algorithm to obtain the MAP estimate of data association variable. We compare the proposed algorithm with track-oriented multiple hypotheses tracking (TOMHT) and PMHT by evaluating the mean optimal sub pattern assignment (MOSPA) performance and computation time for different scenarios.

1 Introduction

Multi-target tracking (MTT) systems find applications in numerous fields such as autonomous vehicular navigation [16], computer vision [14] and the classical radar based tracking applications [2]. The field of MTT has been widely studied and there are many efficient algorithms in practice. However, there still exists room for improvements in several fronts of these systems.

One of the major challenges in multi-target tracking is the data association problem [1, 24]. This problem arises when the sensors in the tracking system return measurements generated by the targets, along with measurements from non-targets, which are termed as clutter in the literature. Besides, the targets are not always detected and the target identity is not available via the measurements. In such scenarios, the problem of identifying the measurement-to-target association constitutes the data association problem. Once this association is determined, estimating the target states is relatively simpler.

The data association problem also becomes computationally difficult when tracking high number of point targets, amidst high clutter density. In these scenarios, several possible data association hypotheses can be formed. The number of ways in which targets can be assigned to measurements at each time instant, while ensuring that a target is assigned to at most one measurement and vice versa, is combinatorial. Further, when analyzed across multiple time instants, the complexity becomes exponential in the number of measurements. Therefore, optimal solution to the data association problem when tracking point targets becomes intractable [2].

Over the years, there have been many algorithms that trade off optimality in order to reduce complexity, when tracking multiple point targets. We briefly discuss few of these algorithms. One of the simplest multi-target tracking algorithm is the global nearest neighbour (GNN). In this algorithm, only the best target-to-measurement association is retained at each time and propagated. In effect, there is only one data association hypothesis at any given time, making the algorithm

computationally simple. Given this data association, state estimation is straightforward. However, this simplicity comes with performance degradation when the target and clutter density is high.

This problem is mitigated to an extent in the commonly implemented MHT algorithm [6, 7, 24] by making a deferred decision. In one of the common implementations of MHT, N-scan pruning based track oriented MHT (TOMHT), several hypotheses are maintained at each time instant. Note that the MHT algorithm runs one Kalman filter (or other Gaussian filters for non-linear models) for each hypothesis retained. The growth in the number of hypotheses is controlled by removing all but one hypothesis, along with its descendants, N time steps back. Increasing the depth N improves the optimality; however, larger values of N result in higher complexity. In particular, the pruning of all but one hypotheses N time steps back is an issue when MHT is employed for batch problems. This is because the estimates of the states N steps before cannot be improved, even with more measurements.

The PMHT algorithm [26, 29] is a batch solution (originally) for target tracking and employs expectation maximization (EM) [12]. In this algorithm, the states are iteratively estimated by implicitly considering several data association hypotheses. This algorithm is computationally simple, as there is only one Kalman filter run per track in each iteration. However, this simplicity is mainly due to the point target assumption being relaxed [10]; when applied to point targets, it results in instances where the same measurement can be assigned to multiple targets. Thus, resulting in coalesced tracks when the targets have very close priors.

The algorithm we present in this paper solves the batch problem of tracking point targets. The algorithm is based on EM, and as a result, it is an iterative algorithm, like PMHT. But, it uses EM to estimate the best data association without relaxing the point target constraints. In each iteration, we have a new data association, implying that we run one Kalman filter per track in each iteration. Thus the algorithm is remarkably simple per iteration, but we have to run several iterations for reasonable results. In our simulations, we found that in most scenarios, the number of iterations need for convergence are less than 10.

As with any other EM algorithm, convergence of the algorithm to a global optimum depends on the initialization. To reduce the sensitiveness of the algorithm to initialization, we have proposed add-on strategies based on homotopy and deterministic annealing strategies. We compare the mean optimal sub pattern assignment (MOSPA) performance and the run time of the proposed algorithm with TOMHT and PMHT algorithms for different scenarios. With the proposed initialization strategies, our algorithm outperforms the version of MHT considered in many challenging scenarios.

In addition to these initialization strategies, we also provide a strategy called track-switching, that improves the convergence, especially when there are closely moving targets. The main idea is to detect close track estimates, force the data association to swap from the point of close proximity and to run the EM algorithm to see if the algorithm converges to a better optimum. For proof-of-concept, we have illustrated this strategy with examples of two closely moving targets in different crossing scenarios, and compared it to MHT. We have observed that the proposed algorithm in tandem with track switching yields improved performance when compared to MHT for these scenarios.

The rest of the paper is organized as follows: In Section 2, we formulate the problem, discuss the challenges and the ways in which the existing algorithms handle them. In Section 3, we derive the EM equations of the proposed algorithm. In Section 4, we present a step-by-step algorithmic description of the algorithm; we also discuss different initialization strategies in this section. Section 5 explains the track-switching strategy. In Section 6, we provide the evaluation of the proposed algorithm by comparing it to TOMHT and PMHT, followed by a summary and possible future works in Section 7.

2 Problem formulation and background

In this section, we present the assumptions we make about the problem, followed by the problem formulation itself and the challenges in arriving at the solution. In the latter part of this section, we also provide a literature overview on the problem.

2.1 Problem formulation

We consider the case of tracking multiple point targets moving in cluttered environment. We assume the number of targets to be fixed and known, denoted as N_T. All the targets are assumed to be present in the observation region at all times. The state vector, $X_{t,i}$, of any target i at time instant t is modelled according to the process model,

$$X_{t,i} = FX_{t-1,i} + V_{t,i}, \tag{1}$$

where F is the state transition matrix and $V_{t,i} \sim \mathcal{N}(0, Q)$ is the process noise with covariance Q. Further, we assume a Gaussian prior at time zero, for all targets, i.e., $X_{0,i}$, $\forall\, i$ with mean $\mu_{0,i}$ and covariance $P_{0,i}$. Targets are assumed to be detected with a detection probability, P_D; when detected, the measurements generated by them are given by

$$Y_{t,i}^{\text{target}} = HX_{t,i} + W_{t,i}, \tag{2}$$

where H is the measurement model and $W_{t,i} \sim \mathcal{N}(0, R)$ is the measurement noise with covariance R. Although we assume linear motion and measurement models in our evaluation, the proposed algorithm can be easily extended to non-linear models. The simplest alternative to incorporate non-linear models is to perform analytical or statistical linearization of the models, and apply the proposed algorithm. The measurement set Y_t at each time can be seen as the union of the set of target measurements (of the detected targets) and the set of clutter detections. Measurements from all time instants are denoted as Y. Clutter measurements are assumed to be uniformly distributed in the observation region of volume V. The number of measurements obtained at time t is denoted M_t. The number of clutter measurements is Poisson distributed with parameter $\beta_c V$, where β_c is the clutter density.

Given the above mentioned assumptions, we are interested in the batch problem of estimating states of all N_T targets at time $t = 1, \ldots, T$, denoted as X, from measurements Y. Also, an estimate of the quality of the state estimates, often captured in a covariance matrix, is desirable. Depending on the tracking application, the estimates can be optimal in ML, MMSE or MAP sense:

$$\hat{X}_{\text{ML}} = \arg\max_X p(Y|X) \tag{3}$$

$$\hat{X}_{\text{MMSE}} = \mathbb{E}[X|Y] \tag{4}$$

$$\hat{X}_{\text{MAP}} = \arg\max_X p(X|Y). \tag{5}$$

The problem we focus in this paper is the MAP estimation of target states.

2.2 Background

In the previous section, we presented the MAP estimation of target states from measurements as the problem of interest. We now explore some of the challenges involved, discuss some relevant existing methods and highlight the weaknesses in them.

In MAP and MMSE estimation problems, the estimates are obtained from the posterior density, $p(X|Y)$. These estimations are tractable, and in many cases straightforward, only when there is no uncertainty in the measurement origin. However, in multi-target tracking problems, the measurement set comprises of measurements from the targets that are detected and also clutter measurements. Further, the measurement origin in the observed set are unknown. This additional uncertainty in the measurement origin must be resolved in order to perform the aforementioned MAP and MMSE estimations. The uncertainty in the measurement origin is commonly handled by introducing the data association variable, which can be used to describe measurement-to-target associations. Using K to represent the data association variable, such that, $K = \{k_{t,i}, \forall t, i\}$, where $k_{t,i} = j$ denotes that target i, at time t is assigned to the measurement $Y_{t,j}$. With the introduction of data association variables, the density of interest becomes $p(X|Y) = \sum_K p(X, K|Y)$. Although the introduction of the data association variable makes it easier to capture the measurement uncertainty, the estimation problem is still intractable due to the huge number of possibilities for K.

Consider the MAP estimation of target states,

$$\hat{X}_{\text{MAP}} = \arg\max_X p(X|Y) = \arg\max_X \sum_K p(X, K|Y), \tag{6}$$

where the optimal point is searched over all the possibilities of K. The massive explosion in the instances of K is because the number of hypotheses increase with the measurements at each scan. That is, the number of hypotheses is exponential in the number of measurements. Thus, sub-optimal approaches are inevitable.

For the linear and Gaussian model assumptions in (1) and (2), each instance of K yields a density $p(X, K = k|Y)$ that is Gaussian, implying that the true posterior density $p(X|Y)$ in (6), which is a sum of such Gaussians over all the values of K is a Gaussian mixture (GM) [5]. All multi-target tracking algorithms resort to making some form of approximation to obtain the target state estimates [2]. We discuss three algorithms below, which are related to the proposed algorithm.

Global nearest neighbour [6] is a simple data association algorithm, widely used in many multi-target tracking applications. GNN approximates the posterior density, which is a GM, as a Gaussian density at each time instant. This is achieved by propagating single hypothesis at each time, thus curbing the growth in the number of hypotheses. By perpetuating a single, high probability hypothesis (at each time instant) translates into hard decision on the target-to-measurements association. As a consequence, if there is a noisy detection and/or if there is high clutter intensity, this algorithm is likely to pick the wrong association, leading to poor performance.

The track-oriented variant of MHT (TOMHT) is the most commonly encountered version of MHT, wherein, multiple data association hypotheses are maintained. For simple cases, when no approximations are made, TOMHT is the optimal data association algorithm. But, for any non-trivial scenarios TOMHT is intractable and approximations are inevitable. Deferring the decision of data association at each time instant to a future time instant, after observing measurements from multiple scans, offers a trade-off between complexity and performance. This is implemented as the N-scan pruning version of TOMHT [15], wherein, multiple data association hypotheses are formed and maintained for N scans. The data association decision at time instant t is deferred to $t + N$, to enable improved decisions after observing measurements from N subsequent time instants. At $t + N$, the branch leading to the best data association hypothesis is traversed back N scans and a hard decision is made at that time instant, discarding away other hypotheses at time t. This implies that the N-scan pruning based TOMHT maintains multiple data association hypotheses for N time instants, and is a single data association hypothesis algorithm for $t - N$. It is evident that the

performance of the algorithm varies with the depth N. When $N = 1$, TOMHT reduces to GNN, due to the instantaneous decisions. In terms of GM, N-scan pruning based TOMHT propagates an approximated GM each time instant, i.e., the true GM is approximated with GM with fewer modes, dependent on the scan depth N.

Both GNN and MHT use pruning, based on hard decisions, to approximate the true GM posterior density. Approximate MAP estimates from GM can also be obtained iteratively, using algorithms such as expectation maximization (EM) [12]. EM is an iterative algorithm used to approximate ML or MAP estimates from models with hidden variables. Probabilistic multiple hypothesis tracker (PMHT) [26, 29] employs EM to iteratively obtain MAP estimates of target states, by treating the data association variables as hidden variables. This results in an algorithm where the MAP estimate of target states and a density over the data association variables are iteratively improved. To reduce the complexity, PMHT makes a rather strong assumption that targets are allowed to generate more than one measurement. This assumption is suitable for tracking extended targets, but is not an accurate one when tracking point targets, where the targets are assumed to generate only one measurement.

The proposed algorithm has interesting connections to the three algorithms discussed above. It is based on EM, and hence is an iterative algorithm like PMHT. But in contrast with PMHT, the proposed algorithm performs MAP estimation of data association variable, by treating the target states as hidden variables. This results in a density over the target states, which can be used to obtain the desired target state estimates. As will be shown in the Section 3, the proposed algorithm uses GNN-like algorithm within its iterations. In each iteration, the proposed algorithm has the possibility of changing the data associations, and can be interpreted to be traversing a different hypothesis; allowing for the notion of multiple hypotheses.

3 EM for multi-target tracking

As established in Section 2, our goal is to estimate the best data association hypothesis K and estimate X, given the best K. In this section, we show how EM can be used to estimate K iteratively from $\Pr\{K|Y\}$. In each iteration, we compute an approximation to the posterior distribution of X, assuming that the current estimate of K is the correct one. In fact, this distribution on X is obtained as part of this optimization problem using EM, which improves with each iteration. Thereon, the state estimates can be obtained from this distribution.

3.1 Introduction to expectation maximization

EM [12] is an iterative technique for approximate MAP estimation, in problems with hidden variables. In our multi-target tracking problem, we are interested in the MAP estimation of the data association variable:

$$\arg\max_{K} \ln \Pr\{K|Y\} = \arg\max_{K} \ln \int p(X, K|Y) dX \tag{7}$$

In applying EM to the MAP problem above, we treat the state variable X as the hidden variable. Starting from this equation, we present a brief introduction to EM as a lower bound on the objective function [3]. Note that there are several other ways to derive the EM equations. In the briefing, we also emphasize on how EM can be used to to estimate the target state X.

EM introduces a distribution in the hidden variable X into the objective function in (7). With this new distribution denoted q_X and using Jensen's inequality, the objective function in (7) can be bounded as follows:

$$\ln \Pr\{K|Y\} \geq \int q_X(X) \ln \frac{P(X, K|Y)}{q_X(X)} dX. \tag{8}$$

The EM iterations maximize this lower bound of the objective function in (8). This new maximization is performed over K and the distribution q_X. In each iteration, for a fixed K, maximization is performed over all the distributions q_X; later, by fixing q_X to be the most recent distribution, maximization is performed over K. It can be shown that the corresponding equations for the iterations are as follows:

$$q_X^{(n)}(X) = p(X|K^{(n)}, Y) \tag{9}$$

$$K^{(n+1)} = \arg\max_K \int q_X^{(n)}(X) \ln \Pr\{K|Y, X\} dX \tag{10}$$

where the superscripts (n) and $(n+1)$ denote the iteration indices. We would like to emphasize that in (9), we have a distribution in X from which we can be used to obtain the state estimates. Equations (9) and (10) can be rewritten to explicitly show the well-known expectation and maximization steps of the EM iterations as follows:

$$K^{(n+1)} = \arg\max_K \mathbb{E}_{p(X|K^{(n)}, Y)} \left[\ln \Pr\{K|Y, X\}\right]. \tag{11}$$

In the next subsection, we derive these iteration equations for the multi-target tracking problem for the model assumptions in Section 2.

3.2 Derivation for data association estimation using EM

In this subsection, we present our major contribution – the EM algorithm for estimating the data association in the multi-target tracking problem. Starting with the EM iteration equations in (9) and (10), we show that for the point target models, these equations simplify to running an RTS smoother [23] and performing a series of 2-D assignments in each iteration.

3.2.1 E step

First, let us consider (9). For point target assumptions, we will soon show that the distribution factorizes over time and target indices. Furthermore, for linear Gaussian assumptions, we will also show that this joint density simplifies to a Gaussian density; the moments of which can be obtained by a forward-backward smoother, such as an RTS smoother.

In (9), given a valid data association sequence $K^{(n)}$, we can show that the joint density $p(X|K^{(n)}, Y)$ factorizes into the marginal densities of the individual targets. For this simplification, we use the assumption that the individual target motions are independent of each other. So, (9) simplifies to $\prod_i p(X_i|K^{(n)}, Y)$, where X_i stands for the state of target i from time 1 to time T. We further simplify this expression using the Markov independence of the states across time, along with the corresponding likelihood factors. Equation (9) now becomes,

$$q_X^{(n)}(X) = \prod_i \prod_t p(X_{t,i}|X_{t-1,i}) p(Y_{t,j}|X_{t,i}, k_{t,i}^{(n)} = j). \tag{12}$$

Note that (12) factorizes across targets. Let us introduce the notation $q_X^{(n)}(X) = \prod_i q_{X_i}^{(n)}(X_i)$ where

$$q_{X_i}^{(n)}(X_i) \triangleq \prod_t p(X_{t,i}|X_{t-1,i})p(Y_{t,j}|X_{t,i}, k_{t,i}^{(n)} = j). \tag{13}$$

This factorization implies that it is enough to handle distribution of each target state individually. In fact, we will show in the second part of this section that obtaining the marginal distributions of the above distribution at each time suffices for the EM iterations. So, we focus on only computing the marginal densities of every target, at each time instant. Let us denote the marginal distribution as $q_{X_{t,i}}^{(n)}(X_{t,i}) \triangleq \int q_{X_i}^{(n)}(X_i)dX_{\setminus t,i}$, where the notation $dX_{\setminus t,i}$ is to indicate that the integration is over the state variable X_i but not $X_{t,i}$. These marginal distributions are typically obtained from (13) by performing smoothing. For linear-Gaussian models in Section 2, one can run an RTS smoother to obtain the Gaussian marginal distribution, which we denote as $\mathcal{N}(X_{t,i}; \mu_{t,i}^{(n)}, P_{t,i}^{(n)})$; in which case, $p(X_{t,i}|X_{t-1,i}) = \mathcal{N}(X_{t,i}; FX_{t-1,i}, Q)$ and $p(Y_{t,j}|X_{t,i}, k_{t,i}^{(n)} = j) = \mathcal{N}(Y_{t,j}, HX_{t,i}, R)$ in (13). To summarize, the equations describing this part of the EM iteration are:

$$q_{X_{t,i}}^{(n)}(X_{t,i}) = \mathcal{N}(X_{t,i}; \mu_{t,i}^{(n)}, P_{t,i}^{(n)}) \tag{14}$$

$$= \int \mathcal{N}(X_{t,i}; FX_{t-1,i}, Q)\mathcal{N}(Y_{t,j}, HX_{t,i}, R)dX_{\setminus t,i}, \tag{15}$$

where the moments $\mu_{t,i}^{(n)}$ and $P_{t,i}^{(n)}$ are obtained using RTS smoother. We will use these marginal densities in the following derivation of (10).

3.2.2 M step

Let us now focus on (10) of the EM iteration. Using the factorization of $\Pr\{K|Y, X\}$, we show that the optimization problem in (10) simplifies to T smaller optimization problems, of finding the best data association at each $t = 1, \ldots, T$. To begin with, consider the term inside the logarithm, $\Pr\{K|Y, X\}$ which is proportional to $p(K, Y|X)$; the proportionality constant depends only on X and Y variables, which are not part of the maximization in (10). For point target model with Poisson clutter assumptions, the following factorization for the term holds (Section 12.3.6 in [18] & [6]):

$$p(K, Y|X) = \prod_{t,i} p(k_{t,i} = j, Y_{t,j}|X_{t,i}). \tag{16}$$

In fact, this factorization holds for most models. Using this factorization, (10) becomes

$$K^{(n+1)} = \arg\max_K \sum_{t,i} \int q_X^{(n)}(X) \times \ln p(k_{t,i} = j, Y_{t,j}|X_{t,i})dX. \tag{17}$$

In the above equation, the integral is with respect to $q_X^{(n)}(X)$ over X, whereas the term $\ln p(k_{t,i} = j, Y_{t,j}|X_{t,i})$ depends only on $X_{t,i}$. In such a case, this integral reduces to an integral that depends only on the marginal distribution over $X_{t,i}$ as follows:

$$K^{(n+1)} = \arg\max_K \sum_{t,i} \int q_{X_{t,i}}^{(n)}(X_{t,i}) \times \ln p(k_{t,i} = j, Y_{t,j}|X_{t,i})dX_{t,i}. \tag{18}$$

For linear-Gaussian assumptions, we have the expression for the marginals $q^{(n)}_{X_{t,i}}(X_{t,i})$ from (14). The other term [6] in the above equation can be shown to be

$$p(k_{t,i} = j, Y_{t,j}|X_{t,i}) \propto \left[\frac{P_D \mathcal{N}(Y_{t,j}, HX_{t,i}, R)}{(1-P_D)\beta_c}\right]^{\mathbf{1}_{k_{t,i}}}, \tag{19}$$

where the indicator function

$$\mathbf{1}_{k_{t,i}} = \begin{cases} 0 & \text{if } k_{t,i} = 0 \\ 1 & \text{otherwise} \end{cases}. \tag{20}$$

Substituting (14) and (19) into (18), we get

$$K^{(n+1)} = \arg\max_K \sum_{t,i} \left[\ln \frac{P_D \mathcal{N}(Y_{t,j}, H\mu^{(n)}_{t,i}, R)}{(1-P_D)\beta_c} - \frac{1}{2}\mathrm{Tr}(P^{(n)}_{t,i} H^T R^{-1} H)\right]^{\mathbf{1}_{k_{t,i}}}, \tag{21}$$

where the notation $\mathrm{Tr}(A)$ stands for the trace of the matrix A.

Now, we can revisit the point target constraints and see its implications on (21). From the point target assumptions, we know that at each time, a target can assigned to at most one measurement and a measurement can be assigned to at most one target. This assumption forces constraints on K, i.e., at each t, $k_{t,i}$ has to be such that the target-to-measurement associations are unique (except for null hypothesis assignment) for different targets i. However, these constraints do not apply across time. This lack of constraints on $k_{t,i}$ across time simplifies (21) to,

$$K^{(n+1)}_t = \arg\max_{K_t} \sum_{i} \left[\ln \frac{P_D \mathcal{N}(Y_{t,j}, H\mu^{(n)}_{t,i}, R)}{(1-P_D)\beta_c} - \frac{1}{2}\mathrm{Tr}(P^{(n)}_{t,i} H^T R^{-1} H)\right]^{\mathbf{1}_{k_{t,i}}}, \tag{22}$$

for $t = 1, \ldots, T$, where the notation K_t denotes the set of target-to-measurement assignments at a particular time t.

The simplification in (22) is remarkable as it allows us to perform optimization independently at each time. And at each time, the optimization problem is a 2-D assignment problem [4]. That is, we are interested in the best target-to-measurement assignment that maximizes the cost function inside arg max in (22). For the tracking community, this assignment is the same problem as in GNN algortihm [6], which is often solved using the modified auction algorithm [11]. Note in GNN, which is a filtering algorithm, the cost function is based on the prediction estimates. Therefore, the cost function for the assignment problem at t, depends on the assignment made at the last time $t-1$ and so on. However, here the cost function depends on the smoothed estimates, so the assignment problems can be solved simultaneously at all time instants. In essence, this part of the EM iteration in (22) requires us to perform a series of auction algorithms, for which the cost functions are based on the smoothed estimates.

In summary, (14) and (22) form the iterations of the proposed EM algorithm strategy for data association estimation. In each iteration, given a data association sequence $K^{(n)}$, we run RTS smoother to compute the marginal densities of the states using (14). Given these marginal densities,

we compute the cost function and run the modified auction algorithm using (22). We run the iterations until convergence. When the algorithm converges, one can obtain the state estimates for trajectories from the mean estimates in (14).

4 Algorithm

In this section, we present a step-by-step description of the proposed algorithm. The algorithm is iterative and thus, it needs a good initialization for the first iteration. In the second half of this section, we also present a few heuristic strategies to handle initialization.

4.1 Description

The proposed algorithm iterations are summarized by (14) and (21). To begin the iterations, we assume we have $K^{(n)}$ for $n = 0$; the choice of this initialization will be discussed in the next subsection. The data association sequence $K^{(n)}$ defines the measurement associations for each target i, and with these associated measurements, an RTS smoother (algorithmic description in Appendix A) is run for each target to obtain the moments $\mu_{t,i}^{(n)}$ and $P_{t,i}^{(n)}$ in (14). With these moments, we compute the cost function term denoted as,

$$C_{ij}(t) \triangleq \ln \frac{P_D \mathcal{N}(Y_{t,j}, H\mu_{t,i}^{(n)}, R)}{(1-P_D)\beta_c} - \frac{1}{2}\mathrm{Tr}(P_{t,i}^{(n)} H^T R^{-1} H), \tag{23}$$

which corresponds to the term inside the summation in (21). We run the modified auction algorithm (in Appendix B) for each time instant with these cost functions $C_{ij}(t)$, which returns the best data association sequence $K^{(n+1)}$ for the next iteration. With this new data association sequence, we are ready to repeat the whole procedure again for the next iteration. A step-by-step description of the algorithm is presented in Algorithm 1.

4.2 Initialization

The proposed EM algorithm approximates the MAP estimate from a Gaussian mixture multi-modal posterior density according to the model description in Section 2. The algorithm performs iterative operations on a multi-modal density; there is no guarantee that the algorithm would converge to the global optimum [19, 28, 30] as this density can have multiple local optima. One way of dealing with the susceptibility to local optima in such scenarios would be to choose good initializations, thereby improving chances of the algorithm converging to the global optimum. There are also strategies that modify the optimization problem such that the algorithms are made less sensitive to the initialization, like the homotopy methods [20] and the deterministic annealing [27]. We discus some of these strategies here.

4.2.1 Choosing good initialization

In the description of the algorithm in the previous subsection, we assumed that we have the initializations $K^{(n)}$ at $n = 0$ for the algorithm to begin. In this subsection, we present a few heuristics to obtain this track initialization. Note that the term 'track initialization' refers to having a good guess of $K^{(0)}$ and does not refer to track births, as is used in tracking community.

Algorithm 1 Multi-target tracking based on EM with hidden state variable

Input: Initialisation $K^{(0)}$
Measurement data Y

while not converged **do**
  E step
  for $i = 1, \ldots, N_T$ **do**
    Set $\tilde{Y}_{t,i} = Y_{t,j}$ such that $K_{t,i}^{(n)} = j$.
    RTS Smoothing: With $\tilde{Y}_{t,i}$ $\forall t$, compute $\mu_{t,i}^{(n)}, P_{t,i}^{(n)}$ $\forall\, t$ using RTS smoother described in Algorithm 4.
  end for
  M step
  for $t = 1, \ldots, T$ **do**
    Cost matrix: Compute cost entries $C_{ij}(t)$, $\forall\, i = 0, \ldots, N_T$, $j = 0, \ldots, M_t$, according to (23).
    2-D assignment With $C_{i,j}(t), \forall\, i, j$ as input, obtain $K_t^{(n+1)}$ using modified auction algorithm described in Algorithm 5.
  end for
  Set $n = n + 1$.
end while

Output: $\mu_{t,i}^{(n)}, P_{t,i}^{(n)}, k_{t,i}^{(n)}$ $\forall\, i, t$.

For the track initialization part of the algorithm, we are interested in $K^{(0)}$, which are sequences of data associations for all the targets. One approach is to use methods like Hough transform [8, 17] which can be used to partition possible sequences of data associations lined up in the scan volume.

Another approach is to obtain the initial data association sequence by running a GNN-based filter with initial estimates $\mu_{0,i}^{(0)}$ of the state at time 0. We have two proposals to obtain these point initialization: uniform initialization, and measurement–based initialization. In the uniform point initialization method, points $\mu_{0,i}^{(0)}$ are uniformly chosen across the observation region. The number of initial points must be large enough, which is dependent on the scan volume and number of targets. In the second method, measurement-based initialization, as the name indicates, initialization is based on measurements. If the probability of detection, $P_D = 1$, all measurements in the first time scan are used as initial points. When $P_D < 1$, measurements from subsequent scans can be used to initialize tracks.

In the aforementioned point initialization methods, numerous initial point estimates of the state vector are used, leading to more tracks than the actual number of targets, N_T, as contenders. The quality of the initial points, $\mu_{0,i}^{(0)}$, are gauged based on the log-likelihood, $\ln p(K, Y|X)$, of the tracks obtained from the initial points. The log-likelihood metric used is closely related to the notion of track score [7] used in target tracking. Unlikely initial points are removed by retaining only the initial points corresponding to the the top N_T log-likelihood values.

4.2.2 Handling sensitivity to initialization

We employ the homotopy–based method to reduce the algorithm's sensitivity to initialization. We briefly discuss this method here.

In the homotopy method [20], the idea is to solve a sequence of optimization problems; in solving each of these problems, we get a step closer to the global optima. This is performed by starting with a modified posterior that is closer to the prior. This can be achieved by setting the measurement noise covariance R to a larger value than its original value, say, by scaling it with a factor α, where $\alpha > 1$. The optimal data association estimate K from this problem is set as the initialization to the next problem. In this new problem, we reduce the scaling α for the measurement covariance one step, and optimize again. This whole procedure is repeated until the scaling α is reduced to 1 and the measurement noise is reduced to its original value R. In this paper, we refer to this method as the inflated R strategy and an algorithmic description is given in Algorithm 2. Regarding choice of α, by trial and error we found that any scaling factor of more than 16 did not help in improving the performance for the scenarios we considered. More on this will be discussed in the results section.

Algorithm 2 EM algorithm with homotopy strategy

Input: Track initialization $K^{(0)}$, measurement data Y, homotopy scaling: α

1: Set $\tilde{R} = R$.
2: Run Algorithm 1 with measurement covariance $R = \alpha\tilde{R}$ to obtain output $K^{(n)}$.
3: **if** $\alpha = 1$ **then**
4: Terminate
5: **else**
6: Reduce α, set $K^{(0)} = K^{(n)}$ and repeat Step 2.
7: **end if**

Output: $\mu_{t,i}^{(n)}, P_{t,i}^{(n)}, k_{t,i}^{(n)} \; \forall \, i, t.$

5 Track switching

In the previous section, homotopy was discussed as one method to tackle sensitivity of the proposed EM-based algorithm to initialization. In this section, we discuss track switching as a possible strategy to further improve the performance of the algorithm.

As the proposed algorithm is based on EM, it is sensitive to initialization. Consequently, the algorithm can converge to a local optimum, depending on the initial data association provided. This is especially aggravted when targets are moving closely. For instance, a pair of crossing targets can be misconstrued to be targets that diverge from the point of crossing, or vice versa; this behaviour of tracking algorithms is discussed as the track repulsion effect in [9]. The two scenarios in Figure 1 (either crossing or repelling after a point) correspond to two different data association hypotheses. In terms of the posterior density, they correspond to two different modes. Depending on the initial data association provided, the algorithm can converge to either of the modes. In many cases, the chosen mode can correspond to the wrong scenario, which degrades performance of the algorithm. If by some means, both the data association hypotheses can be compared, and the hypothesis with the higher posterior probability can be retained, performance in such scenarios can be largely improved.

We present one such strategy to handle scenarios when targets are in close proximity, with the possibility of crossing or repelling from a point. We discuss this method more as a proof-of-concept, and confine the discussion to two targets. The objective of proposing this strategy is to demonstrate that track switching, when performed even with simple techniques, can improve tracking performance of the proposed algorithm. This method we propose can be viewed as deterministic track switching; it has similarities with the stochastic track-switch moves performed in the Markov-chain

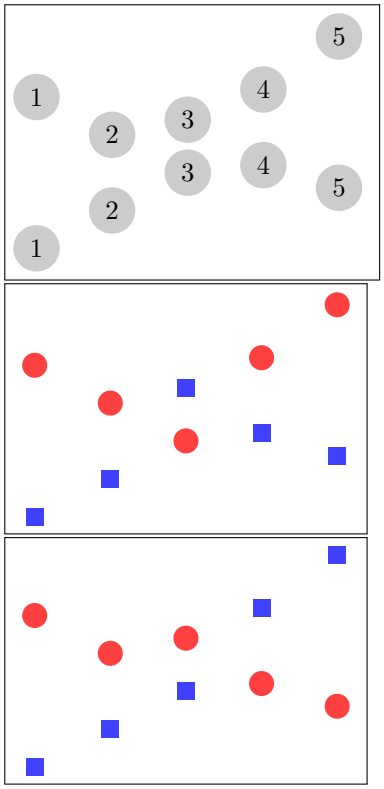


Figure 1: Illustration of possible track switching scenarios. The top figure shows a sequence of measurements observed at different time instants. The second figure depicts the track corresponding to the hypothesis where the tracks are crossing at the third time instant. The third figure depicts the hypothesis that the tracks repel after the third time instant.

Monte Carlo data association algorithm [21].

In order to perform track switching, in potential cases, we assume the estimates from the EM algorithm are available. Given these estimates, the track switching algorithm looks for the possibility that two targets are approaching each other. We deploy a simple method, based on Euclidean distance described next in Section 5.2, to detect if the two targets represent one of the scenarios depicted in Figure 1. And if found to belong to one of the scenarios, track switching procedure is initiated to obtain the second, alternate data association hypothesis. The alternate hypothesis, K', is formed by swapping the incoming data association variable, K, after the detected time of crossing, t_X. The two hypotheses, K and K' are then compared based on their posterior hypothesis, discussed in Section 5.1. The hypothesis with higher probability is retained and the corresponding state estimates are returned as the best estimates. The aforementioned procedure for the deterministic track switching is described in Algorithm 3. The improvement in performance after performing track switching is presented in Section **??**. The details of comparing hypotheses and detecting targets in close proximity are described next.

Algorithm 3 Deterministic track switching for two targets

Input: Target estimates, $\mu_{t,i}^{(n)}$, $P_{t,i}^{(n)}$ $\forall\, t, i = 1, 2$, $K^{(n)}$ from Algorithm 1, measurement data Y.

Set $K = K^{(n)}$ and crossing threshold, σ_X
Detect proximity:
Compute Euclidean distance according to (26).
if Condition in (25) **then**
  Obtain t_X.
  Set $K' = K$
  for $t = t_X, \ldots T$ **do**
    Set $k'_{t,1} = k_{t,2}$, $k'_{t,2} = k_{t,1}$
  end for
  Run Algorithm 1 with K' as initialization to obtain $K'^{(n)}$.
  Compare hypotheses:
  Compute $\Pr\{K^{(n)}|Y\}$ and $\Pr\{K'^{(n)}|Y\}$ according to (24).
  if $\Pr\{K'^{(n)}|Y\} > \Pr\{K^{(n)}|Y\}$ **then**
    for t=1,... T **do**
      $\mu_{t,i}^{(n)} = \mu_{t,i}'^{(n)}$, $P_{t,i}^{(n)} = P_{t,i}'^{(n)}$, $k_{t,i}^{(n)} = k_{t,i}'^{(n)}$, for $i = 1, 2$.
    end for
  end if
end if

Output: $\mu_{t,i}^{(n)}, P_{t,i}^{(n)}, k_{t,i}^{(n)}$ $\forall\, i, t$.

5.1 Comparing alternate hypotheses

The deterministic track switching procedure relies on the reasonable comparison of different data association hypotheses. The best measure to compare any two hypotheses is based on their respective posterior probabilities. It turns out, in deriving the proposed EM algorithm, we have lower bounded the posterior hypothesis probability, $\Pr\{K|Y\}$, in (8). In fact, for the the choice of $q_X(X)$ in (9), it can be shown that the lower bound on the posterior hypothesis probability, is exact [3]. Thus, we have an innate way of comparing hypotheses based on their posterior probabilities. The expression used to obtain the MAP estimate of K, (21) in Section 3, can be used to compare the posterior probabilities of the alternate hypotheses. It is then readily seen that,

$$\Pr\{K^{(n+1)}|Y\} \propto \sum_{t,i} \left[\ln \frac{P_D \mathcal{N}(Y_{t,j}, H\mu_{t,i}^{(n)}, R)}{(1-P_D)\beta_c} - \frac{1}{2}\mathrm{Tr}(P_{t,i}^{(n)} H^T R^{-1} H) \right]^{\mathbf{1}_{k_{t,i}}}. \tag{24}$$

The terms involved in the above expression are inherently computed when the proposed EM algorithm is run, making the computation and comparison of the posterior probabilities of hypotheses all the more simple.

5.2 Detecting targets in close proximity

In this section, a simple strategy based on computing Euclidean distance is described to detect targets in close proximity. Determining if pairs of targets are closely moving is the first step in performing the track switching operation. We detect targets' proximity by comparing the Euclidean distance between the positional estimates at each time instant. Pair of targets are declared close

enough for track switching, if the distance measure between them falls within a predetermined, but tunable threshold. Denoting σ_X as this crossing threshold, track switching is performed if,

$$\min d_t < \sigma_X, t = 2, \ldots, T-1, \tag{25}$$

where d_t is the Euclidean distance between the two targets at t,

$$d_t = \sqrt{(x_{t,1} - x_{t,2})^2 + (y_{t,1} - y_{t,2})^2}. \tag{26}$$

with $(x_{t,1}, y_{t,1})$ and $(x_{t,2}, y_{t,2})$ denoting the position estimates of targets 1 and 2 at t. For obvious reasons, the comparison is futile at first and last time scans. Using this distance measure, the time instant when the two targets are closest is determined, denoted as t_X. We have only used the position estimates to ease the selection of σ_X. Even with this simple threshold, the algorithm has shown substantial improvement in the difficult scenarios under consideration. Employing smarter strategies which can use the velocity and covariance information can improve the detection of targets in close proximity, and hence the overall performance of the algorithm.

Other than using thresholded distance measure to detect targets in close proximity, the measurement-to-target assignment matrix, formed before passing it to the auction algorithm, can also be used. The entries in the assignment matrix, at each time instant, capture all measurement-to-target data associations. Thus, contentions for same measurements between two targets can imply they are in close proximity at that time instant. By basing the decision on several time scans, and also by checking for multiple conflicts within a time scan, a further effective strategy can be formulated to detect targets in close proximity.

6 Simulation and results

In this section we evaluate the performance of the proposed algorithm, along with the impact of homotopy and track switching methods on its performance. Further, the proposed algorithm compared with PMHT and track-oriented MHT for various tracking scenarios. In section 6.1, we first describe the simulation setup and the comparison metrics. In section 6.2, the simulation results are presented.

6.1 Simulations

To evaluate the performance of the proposed EM algorithm, TOMHT and PMHT, we have used the four-dimensional constant-velocity model with additive discrete-time noise, with parameters:

$$X_{t,i} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, F = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$Q = \sigma_v^2 \begin{bmatrix} T_s^4/4 & 0 & T_s^3/2 & 0 \\ 0 & T_s^4/4 & 0 & T_s^3/2 \\ T_s^3/2 & 0 & T_s^2 & 0 \\ 0 & T_s^3/2 & 0 & T_s^2 \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ and } R = \sigma_w^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

Scenario			Position estimates			Velocity estimates		
			TOMHT	Prop.Alg.	PMHT	TOMHT	Prop.Alg.	PMHT
$P_D = 0.9$, $N_T = 20$	$\beta_c =$	0.5	1.43	0.99	2.62	0.51	0.43	0.91
		2.5	1.45	1.65	2.71	0.53	0.60	0.93
		4.5	1.70	1.81	2.95	0.59	0.62	0.99
$P_D = 0.9$, $\beta_c = 1.5$	$N_T =$	10	1.03	1.41	3.86	0.61	0.51	1.49
		20	1.34	1.13	2.52	0.44	0.49	0.88
		30	1.52	0.72	2.11	0.37	0.31	0.74
$N_T = 20$, $\beta_c = 1.5$	$P_D =$	0.8	1.27	1.41	2.25	0.46	0.51	1.03
		0.9	1.15	1.15	2.44	0.42	0.45	0.83
		0.99	1.04	0.89	2.79	0.39	0.38	0.76

Table 1: The table shows RMSE per time scan for position estimates (in m), and velocity estimates (in m/s) based on the MOSPA assignment. The values for each of the scenarios are averaged over 1000 Monte Carlo trials. The values of β_c shown in the table are scaled by 10^4, i.e., in the range of 0.5×10^{-4} to 4.5×10^{-4}.

where $T = 0.5$ s, $\sigma_v = 8m/s^2$ and $\sigma_w = \sqrt{5}$ m for all scenarios. In order to emphasize some interesting beahiour of the proposed algorithm, the following parameters are varied in the simulations:

- $P_D = [0.8, 0.9, 0.99]$, $M_t = 1.5 \times 10^{-4}$, $N_T = 20$
- $\beta_c = [0.5, 2.5, 4.5] \times 10^{-4}$, $P_D = 0.9$, $N_T = 20$
- $N_T = [10, 30, 50, 70, 90]$, $M_t = 1.5 \times 10^{-4}$, $P_D = 0.9$.

All simulation results have been averaged over 1000 Monte Carlo trials. To create dense target scenarios, unless specified otherwise, the targets were generated uniformly from a region of width 80m. We use a scan depth of 3 for the TOMHT algorithm in all the evaluations presented below.

To compare the algorithms, we use Euclidean distance based MOSPA [13, 25] and computation time as the metrics. Computing MOSPA involves calculating the minimum mean squared error between the set of true tracks and estimated tracks, which turns out to be expensive for high number of targets. To do this efficiently, we pose the MOSPA calculation problem as a 2-D assignment problem that attempts to minimize the overall assignment cost (MOSPA) and employ the modified auction algorithm [22]. To compare the computation time, all four algorithms were run on a desktop computer with quad-core processor and 8 gigabyte RAM, running Debian operating system.

6.2 Results

6.2.1 Impact of homotopy method

In order to reduce sensitivity of the proposed EM algorithm to initialization, it was discussed in Section 4.2 that using homotopy–inspired inflation of measurement covariance could be useful. In this section, we present the evaluation that verifies those claims. Figure 2 shows the MOSPA based RMSE position error for the runs of the proposed EM algorithm, with different instantiations of the scaling factor. For the same set of scenarios, the scaling factor α was varied logarithmically from 1 to 16. It is seen that the performance of the algorithm improves with increasing α. In our

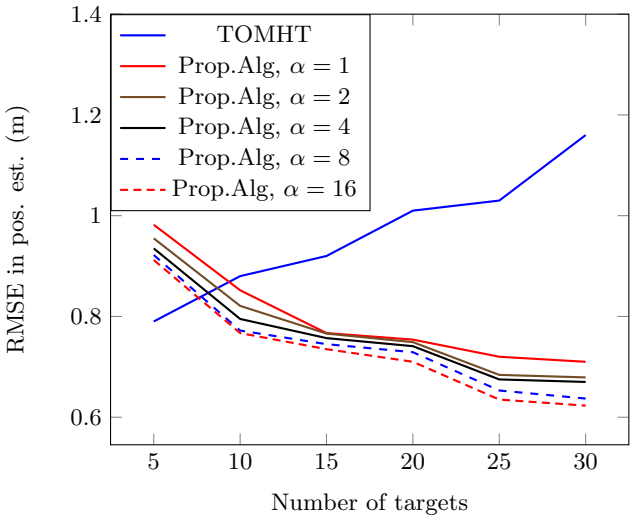


Figure 2: MOSPA based position error comparison for different values of the scaling factor α used in the proposed EM algorithm, as part of the homotopy method.

simulations, we also found that further increase in scaling factor, i.e., $\alpha > 16$ does not improve the performance of the algorithm. The results are also overlaid with TOMHT's performance, for the same scenarios, to give context of the improvement in performance that has been achieved. In the remainder of the simulations, to reduce computation time, we run the proposed algorithm with $\alpha = 8$.

6.2.2 Evaluation of track switching strategy

Evaluation of deterministic track switching, which was presented in Section 5 is discussed here. As mentioned previously, we perform track switching for a pair of targets. To obtain interesting set of results, we have compared the performance of the proposed algorithm with TOMHT for scenarios in which the pairs of targets maneuver at varying angles. Two of the cases are depicted in Figures 3 and 4.

The track switching strategy was evaluated by generating different sets of measurements, for fixed target scenarios at few varying angles. For instance, the tracks in Figure 3 were generated so that the targets could cross sharply. Based on these true targets, several sets of random measurements were generated. The proposed EM algorithm and TOMHT were then run with these measurements, to return their track estimates. By enabling the deterministic track switching, the proposed algorithm was able to resolve the crossing scenario in Figure 3 72% of the trials, whereas, TOMHT was able to resolve the scenario successfully only 35% of the trials. Another scenario, where the targets do not cross, but move closely is depicted in Figure 4. Both Figures 3 and 4, depict the majority behaviour of the algorithms. The improvement in performance of the proposed algorithm has been consistent even with other crossing targets scenarios.

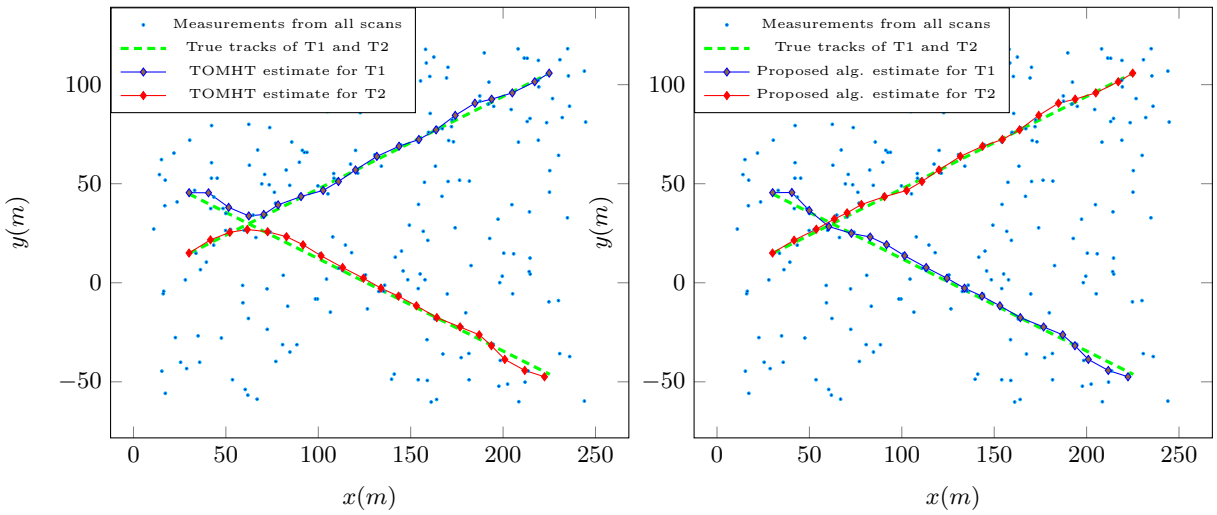


Figure 3: Illustration of an instance, when the true tracks are crossing. TOMHT primarily fails to resolve crossing targets, whereas the proposed algorithm successfully resolves it using track switching, most of the times.

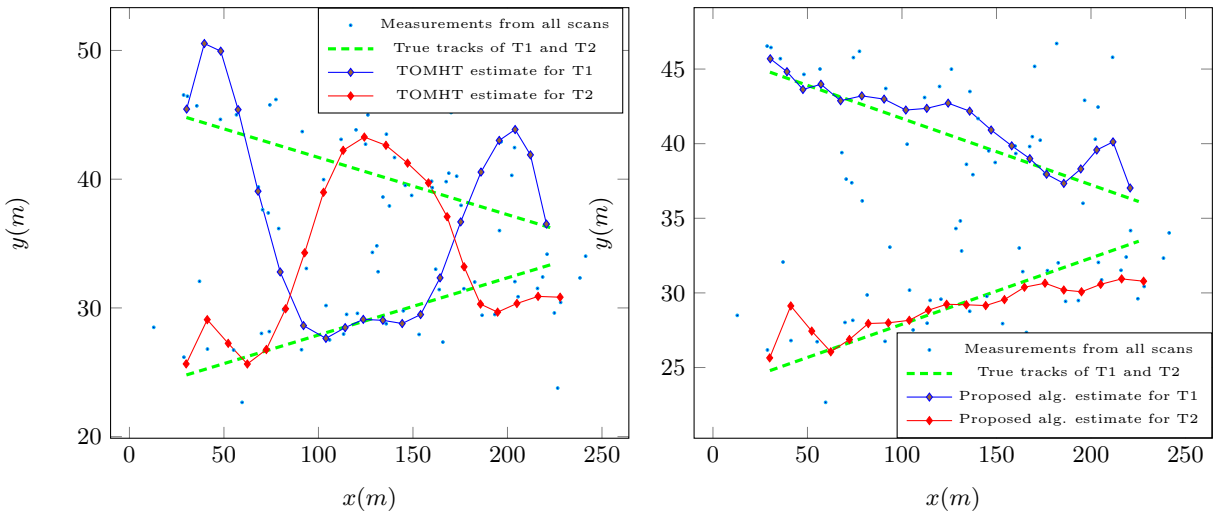


Figure 4: Illustration of an instance, when the true tracks are not crossing, but are moving closely. In this instance, TOMHT primarily outputs tracks that have crossings, whereas the proposed algorithm outputs the non-crossing tracks majority of the trials.

6.2.3 General performance comparison

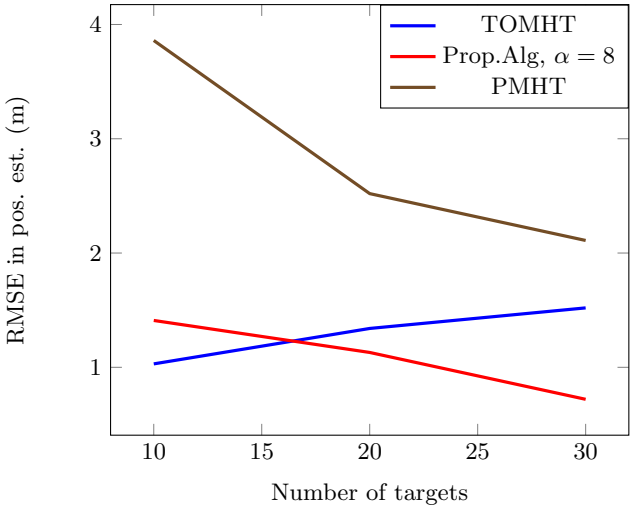


Figure 5: MOSPA position error for different algorithms with varying number of targets.

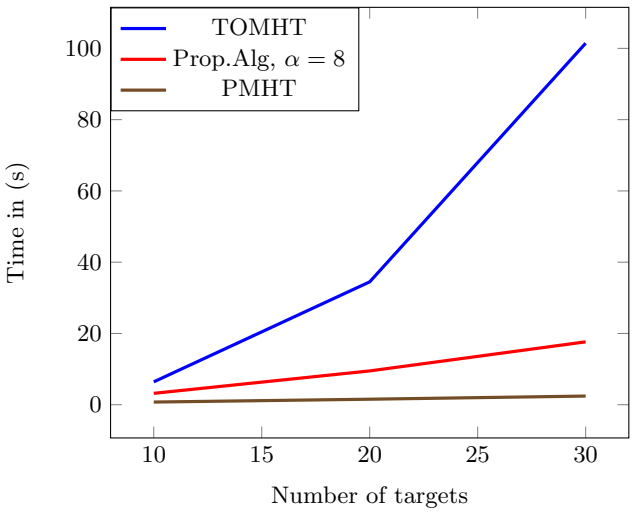


Figure 6: Computation time comparison for different algorithms with varying number of targets.

Results for the comparison of the proposed algorithm with TOMHT and PMHT for varying scenarios is presented in this section. In Table 6, performance of the three algorithms for different scenarios are tabulated. Parameters describing each simulation scenario is obtained from the first two columns. Remaining entries correspond to position and velocity estimate errors, computed based on MOSPA. It is clearly seen that the proposed algorithm performs better than PMHT in all the scenarios we

have considered. Except in sparse situations, the proposed algorithm yields better performance than TOMHT.

Results for the varying number of targets scenario is also repeated in Figure 5. This figure, when seen in conjunction with Figure 6 gives an insight into the significant performance advantage of the proposed algorithm. The proposed algorithm yields better MOSPA performance than TOMHT, but at a slight increase in the computation time.

7 Conclusion and future work

In this paper, we have presented a multi-target tracking algorithm based on expectation maximization as an iterative, batch solution. We have shown that our solution for tracking point targets can be implemented by using simpler soluiton blocks such as smoothing and 2-D assignment. The solution has been presented for the case of known number of targets, and has been compared comprehensively with MHT and PMHT, highlighting the prominent differences between them. Performance of the proposed algorithm has been evaluated for varying number of targets, clutter intensities and probability of detection, and we have shown that our algorithm performs considerably better than PMHT in MOSPA and track loss sense. We have also presented homotopy–based initialization strategy that can alleviate the initializaiton sensitivity for such EM based algorithms. We have described and evaluated deterministic track switching, which further improves the performance of the proposed algorithm.

We see a lot of potential in the proposed algorithm, and foresee many enhancements to it. One natural extension would be to accommodate target birth/death into the solution framework. A coherent approach to track handling could be to model target existence uncertainties as a random variable and estimating it. Another aspect to improving the proposed solution would be to work on an online version of the algorithm. As the algorithm has been shown to be sensitive to initialization, further improvement to initialization techniques, beyond the homotopy–based method would be desirable. Extending track switching to more than two targets, with better thresholding can also be of value.

Appendices

A Smoothing with the RTS smoother

Rauch-Tung-Striebel (RTS) smoother is the ubiquitously used forward-backward smoothing solution. For linear Gaussian models, the forward filtering is performed using a Kalman filter, with the moments calculated as,

$$\mu_{t+1|t} = F\mu_t \tag{27}$$
$$P_{t+1|t} = FP_tF^T + Q \tag{28}$$
$$S_{t+1} = HP_{t+1|t}H^T + R \tag{29}$$
$$K_{t+1} = P_{t+1|t}H^TS_{t+1}^{-1} \tag{30}$$
$$\mu_{t+1} = \mu_{t+1|t} + K_t(y_{t+1} - H\mu_{t+1|t}) \tag{31}$$
$$P_{t+1} = P_{t+1|t} - K_{t+1}S_{t+1}^{-1}K_{t+1}^T \tag{32}$$

Backward smoothing is performed using the moments computed from the forward filter, with the following smoothing equations.

$$
\begin{aligned}
G_t &= P_t F^T P_{t+1|t} && (33)\\
\mu_t^s &= \mu_t + G_t(\mu_{t+1}^s - \mu_{t+1|t}) && (34)\\
P_t^s &= P_t - G_t[P_{t+1|t} - P_{k+1}^s]G_t^T && (35)
\end{aligned}
$$

Algorithm 4 RTS smoothing

Input: Initial estimates μ_0, P_0, and measurement data $y_t \, \forall \, t = 1, \ldots, T$

for all $t = 1, \ldots, T$ **do**

Compute and store the moments $\mu_{t|t-1}$, $P_{t|t-1}$, μ_t, P_t according to (27), (28), (31) and (32), respectively

end for

for all $t = T-1, \ldots, 1$ **do**

Compute the smoothed moments μ_t^s, P_t^s according to (34) and (35),respectively

end for

B Modified auction algorithm

The objective of 2-D assignment is to assign M 'persons' to N 'objects' such that the overall cost of assignment is maximized. Modified auction algorithm gives the optimal solution to the generalized assignment problem [22]. As with real auction processes, the algorithm involves a bidding phase and an assignment phase. During the bidding phase, every unassigned person finds the best object with maximum value, and places a bid for it. In the assignment phase, objects gets assigned to the person with the highest bid. The algorithm performs these two steps iteratively, until all persons have been assigned to an object which maximizes the overall cost.

The bidding phase involves the computation of net value of each object, for a person i,

$$v_{ij} = C_{ij} - p_j \qquad \forall \, j = 0, \ldots, N \tag{36}$$

where, C_{ij} is the cost of assigning person i to object j. p_j is the price vector, initially set to 0. With these new values, the best object j^* offering maximum value to i is found out,

$$
\begin{aligned}
j^* &= \arg\max_j v_{ij} && (37)\\
v_{ij^*} &= \max_j v_{ij} \qquad \forall \, j = 0, \ldots, N && (38)
\end{aligned}
$$

Next, the object with the second best value is found out, as

$$w_{ij^*} = \arg\max_{j \neq j^*} v_{ij} \tag{39}$$

The bid of person i for object j^* is computed,

$$b_{ij^*} = p_{j^*} + v_{ij^*} - w_{ij^*} + \epsilon \tag{40}$$

where, ϵ is a small positive scalar called complementary slackness parameter.

If i' comprises of all the persons who can place a bid for object j^*, assignment phase involves assigning the object to the highest bidder, $j^* \rightarrow i^*$ and increasing the price of the object to the highest bid, performed as,

$$i^* = \arg\max_{i'} b_{i'j^*} \tag{41}$$

$$p_j := \max_{i'} b_{i'j^*} \tag{42}$$

Algorithm 5 Modified auction algorithm

Input: ϵ, the slackness parameter,
Cost matrix C, with entries $C_{i,j}$, $\forall\, i = 0, \ldots, M$,
$j = 0, \ldots, N$.

Initialize: $q = [1, \ldots, M]$,
$p_j = 0$, $\forall\, j = 0, \ldots, N$,
Assignment matrix, $K = [0, \ldots, 0]_{1 \times M}$
$i' = 0$.
while q is non-empty **do**
  Bidding phase
  Load i from queue, $i = q(1)$.
  Compute net value, v_{ij}, $\forall\, j = 0, \ldots, N$, according to (36).
  Find the best object j^* for i according to (37), with value v_{i,j^*} according to (38).
  Compute i's bid for j^*, b_{ij^*}, according to equation (40).
  Assignment phase
  Find out i', along with their bids, $b_{i'j^*}$
  Choose the highest bidder i^* from i', according to (41).
  if i^*==i **then**
    Assign i^* to j^*, i.e., $K(i^*) = j^*$.
    Append j^*'s previously assigned bidder to q.
    Remove i^* from q.
  end if
  if i^* is non-zero **then**
    Update object prices, p_j, according to (42).
  end if
end while

Output: K with optimal assignments

References

[1] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 618–626, 1978.

[2] ——, "Multitarget-multisensor tracking: Advanced applications," *Norwood, MA, Artech House, 1990, 391 p.*, vol. 1, 1990.

[3] M. J. Beal, "Variational algorithms for approximate bayesian inference," Ph.D. dissertation, University of London, 2003.

[4] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Annals of operations research*, vol. 14, no. 1, pp. 105–123, 1988.

[5] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

[6] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood: Artech House, 1999.

[7] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, 2004.

[8] B. D. Carlson, E. D. Evans, and S. L. Wilson, "Search radar detection and track with the hough transform. iii. detection performance with binary integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 116–125, 1994.

[9] S. Coraluppi, C. Carthel, P. Willett, M. Dingboe, O. O. Neill, and T. Luginbuhl, "The track repulsion effect in automatic tracking," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. IEEE, 2009, pp. 2225–2230.

[10] D. F. Crouse, M. Guerriero, and P. Willett, "A critical look at the PMHT," *Journal of Advances in Information Fusion*, vol. 4, no. 2, pp. 93–116, 2009.

[11] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A generalized SD assignment algorithm for multisensor-multitarget state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 523–538, 1997.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[13] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in *International Conference on Information Fusion*. IEEE, 2010, pp. 1–8.

[14] D. Koller, J. Weber, and J. Malik, *Robust multiple car tracking with occlusion reasoning*. Springer, 1994.

[15] T. Kurien, M. Liggins *et al.*, "Report-to-target assignment in multisensor multitarget tracking," in *IEEE Conference on Decision and Control*. IEEE, 1988, pp. 2484–2488.

[16] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*. Ieee, 1991, pp. 1442–1447.

[17] H. Leung, Z. Hu, and M. Blanchette, "Evaluation of multiple target track initiation techniques in real radar tracking environments," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 143, no. 4, pp. 246–254, 1996.

[18] R. P. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[19] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[20] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[21] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *IEEE Conference on Decision and Control*, vol. 1. IEEE, 2004.

[22] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Transactions on Automatic Control*, vol. 37, no. 2, pp. 198–213, 1992.

[23] H. E. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[24] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[25] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[26] R. L. Streit and T. E. Luginbuhl, "Probabilistic multi-hypothesis tracking," DTIC Document, Tech. Rep., 1995.

[27] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Neural Networks*, vol. 11, no. 2, pp. 271–282, 1998.

[28] B. Wang, D. Titterington *et al.*, "Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model," *Bayesian Analysis*, vol. 1, no. 3, pp. 625–650, 2006.

[29] P. Willett, Y. Ruan, and R. Streit, "PMHT: Problems and some solutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 738–754, 2002.

[30] C. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.