Norwegian University of Science and Technology

TTK4550 Project report

Multihypothesis data association in multitarget tracking

Supervisor:
Edmund F. Brekke
Cosupervisors:
Lars-Christian N. Tokle
Erik F. Wilthil

Student:
Odin A. Severinsen

19th December 2022

Abstract

This report explores estimation of association hypothesis marginals in multitarget tracking when in a multihypothesis setting. The work builds upon the recent results in multitarget tracking where the algorithm loopy belief propagation (LBP) for single-hypothesis cases has seen much success. There are two contributions in this report. The first is a novel factor graph representation of the joint multihypothesis association hypothesis posterior. The second contribution are two algorithms that both are based on loopy belief propagation. The first algorithm uses total probability in conjunction with hypothesis-conditionend LBP and estimation of associated likelihood, and is called LBP-PHD. The second method is the main, theoretical result in the report and is an LBP algorithm running directly on the full multihypothesis association graph with novel, specialized message definitions that are derived in this report and efficient to compute and store in memory, and is called MH-LBP.

Results show that both algorithms perform well with high correlation with the exact marginals for the majority of the cases. The MH-LBP algorithm shows less variance than LBP-PHD, but results also show that much error can be attributed to the estimated likelihood, which shows promise for LBP-PHD if a better likelihood estimation method can be derived. MH-LBP fails to converge for exactly one case, and it is speculated whether this is related to the underlying Bethe free energy of the multihypotheis factor graph.

Sammendrag

Denne rapporten utforsker estimering av assosiasjons-hypotese-marginaler i målfølging av flere mål under målfølging med flere hypoteser. Arbeidet bygger på resultater i målfølging av flere mål hvor algoritmen "loopy belief propagation" (LBP) i enkelhypotesetilfeller har vært svært suksessfull. Det er to hovedforskningsbidrag i denne rapporten. Det første bidraget er en ny faktorgrafrepresentasjon av simultan-multihypotese-assosiasjons-posterioren. Det andre bidraget er to algoritmer for marginalestimering som begge er basert på LBP. Den første algortimen bruker total sannsynlighet sammen med hypotesebetinget LBP og estimering av assosiert målingssannsynlighettetthet, og kalles LBP-PHD. Den andre er hovedteoriresultatet i rapporten og er en LBP-algoritme som kjører direkte på den fulle multihypotese-assosiasjonsgrafen med nye, spesialiserte meldingsdefinisjoner som er utledet i denne rapporten og som er effektive å kjøre og lagre i minnet, og kalles MH-LBP.

Resultater viser at begge algoritmer estimerer med god ytelse og høy korrelasjon med de eksakte marginalene i flertallet av tilfellene. MH-LBP-algoritmen har mindre varians enn LBP-PHD, men resultater viser også at mye feil kommer fra estimering av målingssannsynlighettettheten, noe som viser potensiale for LBP-PHD dersom en mer nøyaktig metode for å estimere denne kan utledes. MH-LBP klarer ikke å konvergere i nøyaktig ett tilfelle, og det spekuleres i om dette kan være relatert til “Bethe free energy”-funksjonen til multihypotese-faktorgrafen

Preface

This report is the result of the specialization project TTK4550 at the department of Engineering Cybernetics fall 2016. The work will continue during the master thesis in the spring of 2023.

This project has been in cooperation with Zeabuz AS, and would like to thank you for allowing me a part-time job there this fall. In particular, I would like to thank my cosupervisor at Zeabuz, Erik Wilthil.

I would also like to thank my supervisor Edmund F. Brekke together with my cosupervisor at NTNU, Lars-Christian Ness Tokle. You have both been available at all hours of the day for discussions, and my growth during this project would not be possible without your guidance.

Odin Aleksander Severinsen

Trondheim, December 2022

Table of Contents

Abstract i

Sammendrag ii

Preface iii

Table of Contents vii

List of Tables viii

List of Figures x

List of Abbreviations xi

I Introduction and preliminaries

1 Introduction 1

1.1 Related work . . . 2

1.2 Contribution and motivation . . . 3

1.3 Outline of report . . . 3

2 Factor graphs 5

2.1 Variable elimination . . . 5

2.1.1 Elimination order and dynamic programming . . . 6

2.2 Encoding structure in densities with factor graphs . . . 8

2.3 Belief propagation . . . 8

2.3.1 The sequential formulation . . . 9

2.3.2 The parallel formulation . . . 10

2.4 Loopy belief propagation . . . 10

2.4.1 LBP's relation to variational inference - The Bethe free energy 11

2.4.2 Research results on properties of loopy belief propagation . 13
2.4.3 Determining convergence of loopy belief propagation . . . 13

3 Data association in multitarget tracking 14
3.1 Multitarget tracking models and assumptions for data association 14
3.1.1 Track state estimation 14
3.1.2 Overview of MTT specific models and concepts 16
3.1.3 Gating of measurements 19
3.2 Hypothesis generation in MTT 20
3.2.1 Hypothesis generation in JPDA 20
3.2.2 Hypothesis generation in MHT 24
3.3 Marginalization of joint association hypothesis posterior 31
3.3.1 Explicit hypothesis enumeration 31
3.3.2 Finding the M best hypotheses with Murty's method 34

II Multihypothesis data association in multitarget tracking

4 Approximating the association marginals 36
4.1 Factor graph representation of joint multihypothesis association hypothesis posterior 36
4.2 Hypothesis-conditioned loopy belief propagation with probability hypothesis density approximation likelihood 39
4.3 Multihypothesis loopy belief propagation 40
4.3.1 Algorithmic complexity 49

III Simulation results

5 Simulation of multitarget tracking scenarios 52
5.1 Overview of track clusters statistics used for testing 52
5.1.1 Implications of number of competing tracks from a graphical point of view 54
5.2 The methods compared 54

6 Results and discussion 56
6.1 Signed marginal error and bias 56
6.2 Survival function over association errors 57

6.3 Normalization constant accuracy 58
6.4 Correlation between approximate and exact marginals 58
6.5 Failed convergence of MH-LBP 61

IV Closing remarks

7 Conclusion **66**
7.1 Future work 67

V Appendices

Bibliography **70**

List of Tables

4.1 Message types in association graph. 42

6.1 Summary of signed errors statistics for the different inference methods tested on the simulated dataset. 56

List of Figures

2.1 Example factor graph of the function $f(x_1, x_2, x_3, x_4, x_5) = f_1(x_1, x_2)f_2(x_2, x_3, x_4, x_5)$. 6

2.2 Figures comparing the two message definitions what are used when doing belief propagation on a factor graph. 9

3.1 Example of a simple association problem in MTT and how a measurement-oriented hypothesis tree is constructed from a parent hypothesis $\theta^l_{1:k-1}$ into eleven new child hypotheses $\theta^{1:11}_{1:k}$ in the same manner as in [40]. Each layer in the tree corresponds to different associations to the same measurement, indicated by the dashed lines, such that the layer index together with the index in the node corresponds to an association. The index 0 is used to indicate misdetection while the indices 3 and 4 refer to the new track index that is initialized in the unassociated measurement. A full hypothesis can be retrieved by traversing the tree from the root to a leaf, where each branch is a hypothesis. Note also that we chose to enumerate the child hypotheses by $1, 2, \ldots, 11$ for convenience, but that it of course is not necessary in general. 26

3.2 Hypothesis tree based on the scenario in Figure 3.1 but where we only enumerate new targets. Comparing the number of leaf nodes in the two examples, the benefits of using the combined clutter and new target declaration is evident, as this restricts the growth of the hypothesis tree. 29

4.1 A toy example with three tracks a^1, a^2 and a^3 and two measurements b^1 and b^2. 39

4.2 Simplified illustraion of message directions in association graph. 42

5.1 Histograms over number of tracks, number of. 53

5.2 Two different association cases. The left case is a hard case, as the two tracks a^1 and a^2 are both competing for the measurements b^1 and b^2 with close to equal detection likelihood, making two strong cycles in the graph. In the right case, however, a^2 is the only track that gates b^2 with a weak link, corresponding to low detection likelihood, to b^1, such that graph more resembles a tree, making the LBP approximation closer to exact. 54

6.1 Histogram of the signed errors between MH-LBP and hypothesis-conditioned LBP with PHD approximation. Note that the y-axis is logarithmic . . . 57
6.2 Survival function for different errors. Note that both the y-axis and x-axis are logarithmic. The vertical line at $x = 0$ is due to all the marginals that estimated with zero error. 59
6.3 Correlation plot between estimated normalization constant and true normalization constant with logarithmic scaling. 60
6.4 Heatmap showing correlation between MH-LBP marginals and exact marginals. Note that the colors are logarithmic. 62
6.5 Oscillations of the different sigma messages of the tracks in the cluster where MH-LBP did not converge, together with the cluster-conditioned reward matrix. Note that only 30 iterations are plotted. 63
6.6 Convergence of messages after increasing the misdetection probabilities. The logarithm of the misdetection probabilities were increased by 2.5, effectively multiplying them by $e^{2.5}$. Note that 150 iterations are plotted to capture the convergence. 64

List of Abbreviations

BP Belief propagation

FISST Finite set statistics

IPDA Integrated probabilistic data association

JIPDA Joint integrated probabilistic data association
JPDA Joint probabilistic data association

LBP Loopy belief propagation
LBP-PHD Hypothesis-conditioned loopy belief propagation with PHD approximation likelihood

MH-LBP Multihypothesis loopy belief propagation
MHT Multiple hypothesis tracker
MTT Multitarget tracking

NIS Normalized innovation squared

PDAF Probabilistic data association filter
PDF Probability density function
PHD Probability hypothesis density
PMBM Poisson multi-Bernoulli mixture

SNR Signal-to-noise-ratio
STT Single-target tracking

I

Introduction and preliminaries

1 | Introduction

A critical component to any autonomous system is the ability to understand its surroundings, also called *situational awareness*. In an automotive setting, this means e.g. to detect and act based on other surrounding cars. In a maritime setting, this might be an autonomous ferry that wishes to cross a canal busy with traffic. The problem of detection, estimation and prediction of the state of external vessels in this sense is referred to as *target tracking*, and is commonly divided into *Single-target tracking* (STT) and *Multitarget tracking* (MTT).

Single-target tracking is, as the name suggests, the objective of tracking a single target. Although far from a trivial task, the assumption that there is only one target makes what is called *data association*, i.e., to associate measurements of the target, usually as a position, significantly easier. For this to give the “correct” estimate, however, no other targets can be competing for the received measurements. In other words, it is possible to run multiple single-target trackers in parallel as long as the targets are sufficiently far from each other to not interfere. The more complex problem of multitarget tracking allows for multiple targets to compete for the same measurements, and makes data association considerably more complex. In the following text, only multitarget tracking will be considered.

In order to do data association in multitarget tracking, one needs to build what is called *association hypotheses* that associates measurements to the *track* of a given target, which refers to the estimated trajectory that the true target follows. As time progresses, the number of possible hypotheses that can be made from the measurements that one receives each time step grows exponentially. This makes naive enumeration of all of them infeasible from a computational perspective. By introducing some assumptions about what associations are valid to make, a particular structure in the data association problem is revealed, which is the key result and novelty that the following report will investigate. By combining the particular structure with some approximations, solutions close to the true solution can be calculated with considerably less computations, which allows for multitarget tracking data association for online purposes.

1.1 Related work

The use of probabilistic graphs for inference in robotics and autonomous systems has seen rapid increase the last decades. The earliest work are on two representations called *Markov random fields* [1]–[3] and *Bayesian networks* [4], [5], where the main difference lies in their representation abilities of the underlying probability density. A major breakthrough in the usefulness of probabilistic graph modelling can be traced back to the work of Pearl [4] which first described the efficient inference method *Belief propagation* (BP) on probabilistic graphs with tree structure which introduces the concept of *message passing* between nodes. The Kalman smoother, from the famous and widely used Kalman filter [6], can be interpreted as doing BP on these types of probabilistic trees under Markov and linear Gaussian assumptions about the state and measurements

The present work takes the more common approach in multitarget tracking of using *factor graphs*, a third graph representation first presented by Kschischang et. al in [7], to efficiently approximate a solution to the data association problem. Early work in this field was done by Chen, Cetin et. al in [8]–[11] where they use message passing for find the optimal association hypothesis by using the *max-product* algorithm, a close relative of BP that finds the argmax of a joint distribution instead of marginalizing it.

More recent work by Williams et. al in [12], [13] augments the data association problem by overparameterizing of association variables which allows for formulating a bipartite matching graph and applies *Loopy belief propagation* (LBP) to efficiently and quickly compute approximate association marginals that can be used in a MTT filter, such as in [14]. Together with Vontobel in [15] they prove that this graphical representation exhibits certain properties that guarantees convergence of LBP, a particularly desirable property. In one of their latest work [16] they do approximate marginalization on an association graph similar to the one in [13] generalized for multiscan. They derive a BP-like algorithm based on a convex approximation to the exact, nonconvex Bethe free energy of the graph for better and more robust performance.

In the work by Meyer, Braca et. al [17] they embedded the data association method presented in [13] in a factor graph representation of the joint track state posterior in a multisensor setting and uses LBP to approximate the marginal track state posteriors. They later extend this method with estimation of unknown, time-varying model parameters [18] and the presence of an unknown number of targets [19].

Lastly, in the maritime setting, Gaglione et al. proposes a method for multisensor-multitarget tracking by constructing a suitably devised factor graph and use LBP for approximate inference in [20]. In [21] the same authors uses BP to perform data fusion of radar and AIS (Automatic Identification System) data.

1.2 Contribution and motivation

A common denominator in all the applications of factor graphs for data association mentioned in Chapter 1.1 is that they are used in a *single-hypothesis* setting. The novelty in the present work is to propose two different approaches that generalize the approximate data association method presented in [13] to a *multihypothesis* setting.

The motivation for this is twofold. The first is the computational benefits, as LBP has been shown to compute good approximations to the association marginals in the aforementioned work with a fraction of the computations needed for an exact solution. The second reason is for *track management* in the MTT filter *Poisson multi-Bernoulli mixture* (PMBM) that was first presented in [14] by Williams et. al. Inside the PMBM framework, new tracks are initialized for every measurement that is not associated to a new track, which over time means the number of tracks to estimate is unbounded without any pruning procedure. For a single-hypothesis tracking scenario Williams proposes in a previous work [22] the concept of *recycling*, which means to return low-quality tracks, i.e. tracks with low existence probability, into the Poisson component for undiscovered targets. By generalizing the method in [13] for multihypothesis scenarios we are able to achieve the same for the multihypothesis case.

1.3 Outline of report

The report is structured as follows. First, the theory of *factor graphs* is introduced in Chapter 2 that make up the tools that are later used for approximate inference of association marginals. In Chapter 3, MTT modeling assumptions are introduced that makes data association doable, together with the derivation of different joint association hypothesis posteriors and how to marginalize them. The main chapter of this report is Chapter 4, which is where the main contributions and novel work is presented. Here, the actual problem is described, together with a derivation of the proposed solution. Lastly, results of the given method are presented in Chapter 6, followed by a conclusion and further work in Chapter 7.

2 | Factor graphs

Factor graphs, first described in [7], is a particular *bipartite graph* consisting of *variable nodes* and *factor nodes* where edges are only between variables and factors. A factor graph describes a function $f(x_{\mathcal{V}})$ which can be *factorized* as

$$f(x_{\mathcal{V}}) = \prod_a f_a(x_{\mathsf{N}(a)}) \tag{2.1}$$

where $\mathcal{V}$ denotes the *variable index set* of the graph such that $x_{\mathcal{V}}$ indicates all variables x_i, $i \in \mathcal{V}$, of f and where $f_a(x_{\mathsf{N}(a)})$ is a *factor* of f with $\mathsf{N}(a)$ indicating all neighbors of node a such that $x_{\mathsf{N}(a)}$ indicates all neighboring variable nodes of f_a.

Factor graphs are useful data structures that have seen use in fields like communication theory for decoding purposes [23]. We will here concern ourselves with how they can model a *Probability density function* (PDF). For this purpose, factor graphs are only one possible graphical representation, where the two other ones are *Markov random fields* and *Bayesian networks*. Describing these representations are outside the scope of this text, but the reader is instead referred to references like [24], [25]. For the purpose of this text, it suffices to say that factor graphs are particularly useful for describing functions where the variables *locally* depend on each other, i.e. where the factors of a complicated, global function are simpler and more local [7]. We will see that this is often a modelling assumption we make about PDFs to facilitate efficient inference algorithms that operate locally.

2.1 Variable elimination

Before proceeding with more specialized algorithms for inference on factor graphs, we will first discuss the most general approach, called *variable elimination*. The name "variable elimination", as the name suggests, comes from the fact that we eliminate variables from the graph by *marginalization* of the underlying PDF $p(x_{\mathcal{V}}) \propto f(x_{1:n})$.

The marginal distribution $p(x_\nu)$ for a subset ν of the variables in $\mathcal{V}$, $\nu \subset \mathcal{V}$, can be retrieved by marginalizing out all other variables of the density, denoted by $\bar{\nu} = \mathcal{V} \setminus \nu$ and with $v_i \in \bar{\nu}, i = \{1, \ \dots \ , |\bar{\nu}|\}$,

$$\begin{aligned} p(x_\nu) &= \sum_{\bar{\nu}} p(x_\mathcal{V}) \\ &= \sum_{x_{v_1}} \sum_{x_{v_2}} \cdots \sum_{x_{v_{|\bar{\nu}|}}} p(x_\mathcal{V}). \end{aligned} \tag{2.2}$$

The sum in (2.2) might at first look harmless and straight-forward to compute. The tractability of the sum, however, heavily depends on *the order the variables are summed out, called the elimination order, dynamic programming* and *the structure of the graph.* We will here consider an example to build intuition for the two former points, while the latter point will be inspected for a concrete case later in Chapter 2.3.

2.1.1 Elimination order and dynamic programming

Consider the factor graph in Figure 2.1 which represents the probability density

$$p(x_1, x_2, x_3, x_4, x_5) \propto f(x_1, x_2, x_3, x_4, x_5) \tag{2.3}$$

$$= f_a(x_1, x_2) f_b(x_2, x_3, x_4, x_5). \tag{2.4}$$

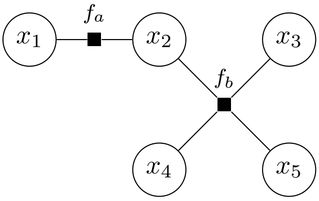


Figure 2.1: *Example factor graph of the function* $f(x_1, x_2, x_3, x_4, x_5) = f_1(x_1, x_2) f_2(x_2, x_3, x_4, x_5)$.

Suppose that all variables of f can take values from a discrete and finite set denoted by $\mathcal{X}$ and that we wish to marginalize out x_2 and x_4, i.e. *eliminate* x_2 and x_4. The procedure looks like

$$\tilde{f}(x_1, x_3, x_5) = \sum_{x_2, x_4} f(x_1, x_2, x_3, x_4, x_5) \tag{2.5}$$

$$= \sum_{x_2, x_4} f_a(x_1, x_2) f_b(x_2, x_3, x_4, x_5). \tag{2.6}$$

Doing this naively involves $\mathcal{O}(|\mathcal{X}|^2)$ summations that has to be repeated $|\mathcal{X}|$ times for each variable in the marginalized function $\tilde{f}$, which totals in $\mathcal{O}(|\mathcal{X}|^5)$ computations. We can improve the complexity with *dynamic programming* by instead carefully do one sum at a time and *store temporary computations*. This entails choosing an order to do the sums – the *elimination order*.

If we eliminate x_2 first and then x_4, the resulting computations are

$$\begin{aligned}
\sum_{x_2,x_4} f(x_1,x_2,x_3,x_4,x_5) &= \sum_{x_4}\sum_{x_2} f_1(x_1,x_2)f_2(x_2,x_3,x_4,x_5) && (2.7)\\
&= \sum_{x_4}\underbrace{\sum_{x_2} f_1(x_1,x_2)f_2(x_2,x_3,x_4,x_5)}_{\tilde{f}_1(x_1,x_3,x_4,x_5)} && (2.8)\\
&= \sum_{x_4} \tilde{f}_1(x_1,x_3,x_4,x_5) && (2.9)\\
&= \tilde{f}(x_1,x_3,x_5). && (2.10)
\end{aligned}$$

The largest computation is in (2.8), where we build a table $\tilde{f}_1$ of size $|\mathcal{X}|^4$ by doing a summation of size $\mathcal{O}(|\mathcal{X}|)$ for each entry, which in total makes the marginalization $\mathcal{O}(|\mathcal{X}|^5)$, the same as for naive marginalization. In other words, for the elimination order x_4, x_2 we gain nothing. We can, however, do better. If we instead eliminate x_4 and then x_2, the computations become

$$\begin{aligned}
\sum_{x_2,x_4} f(x_1,x_2,x_3,x_4,x_5) &= \sum_{x_2}\sum_{x_4} f_a(x_1,x_2)f_b(x_2,x_3,x_4,x_5) && (2.11)\\
&= \sum_{x_2} f_a(x_1,x_2)\underbrace{\sum_{x_4} f_b(x_2,x_3,x_4,x_5)}_{\tilde{f}_2(x_2,x_3,x_5)} && (2.12)\\
&= \sum_{x_2} f_a(x_1,x_2)\tilde{f}_2(x_2,x_3,x_5) && (2.13)\\
&= \tilde{f}(x_1,x_3,x_5) && (2.14)
\end{aligned}$$

where we moved $f_a(x_1,x_2)$ out of the sum in (2.12) as it is constant with respect to x_4. In this case, the largest computation is in (2.12), where we build a table $\tilde{f}_2$ of size $|\mathcal{X}|^3$ by doing a summation of size $\mathcal{O}(|\mathcal{X}|)$ for each entry, which in total makes the marginalization $\mathcal{O}(|\mathcal{X}|^4)$, improving the computation.

As the example above shows, the order of elimination has an impact on the computational complexity. This begs the question whether we can find an optimal elimination order and also how efficient doing elimination in this order is. Unfortunately, in the general case, finding an optimal elimination order is NP-complete, and even when using this order the elimination procedure is NP-hard [24]. Fortunately, all hope is not lost. The remaining discussions in this chapter will explore what structure we require to be able to do efficient inference and how we might achieve similar performance even when this is violated.

2.2 Encoding structure in densities with factor graphs

As we saw in Chapter 2.1, in the general case inference on factor graphs is NP-complete, and infeasible in practice. Thus, we need to invoke additional model assumptions about the *conditional independence* between the variables. As an example, consider again the joint distribution in (2.3). By naive application of the chain rule of probability, we can factorize it as

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2|x_1)p(x_3, x_4, x_5|x_1, x_2). \tag{2.15}$$

Comparing (2.15) with the factorization in (2.4) and matching factors, we recognize that

$$f_a(x_1, x_2) \propto p(x_1)p(x_2|x_1) \tag{2.16}$$

$$f_b(x_2, x_3, x_4, x_4) \propto p(x_3, x_4, x_5|x_1, x_2) \tag{2.17}$$

which reveals that $p(x_3, x_4, x_5|x_1, x_2) = p(x_3, x_4, x_5|x_2)$ which implies that x_3, x_4, x_5 are conditionally independent of x_1 given x_2, denoted by $x_3, x_4, x_5 \perp\!\!\!\perp x_1 \mid x_2$. In this case, making the assumption turns the factor graph into a *factor tree*, i.e. there are no loops in the factor graph. This has profound consequences that are explored in Chapter 2.3.

2.3 Belief propagation

The inference algorithm BP was first described in [4] and later in [7] for factor graphs, and is an algorithm for doing inference on probabilistic graphs with *tree structure* that exploits the structure of the graph for major efficiency improvements.

The objective of BP is to efficiently compute the marginals of each variable x_i in the graph. In other words, given a distribution $p(x_1, x_2, \ldots, x_n)$ we seek each marginal $p(x_i), i = 1, \ldots, n$. When we marginalized the distribution in Figure 2.1 we observed that choosing the elimination order x_4, x_2 we were able to reduce the computational complexity. This is no coincidence. In fact, the factor graph in Figure 2.1 is indeed a *factor tree*, satisfying the requirement of BP, and so we can attribute the performance gain to the fact that we started in a *leaf node* of the graph.

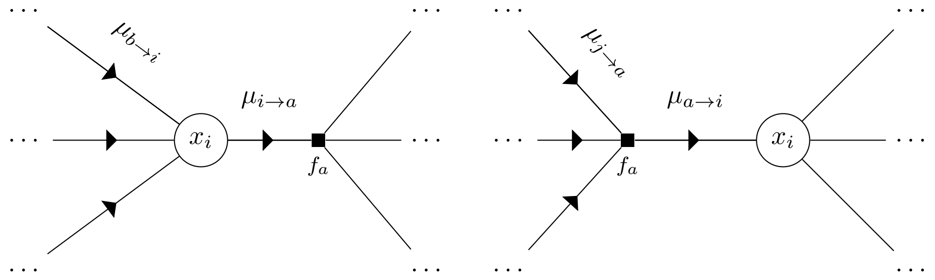
2.3.1 The sequential formulation

This takes us to how BP operates. To use the example above as a stepping stone towards the general algorithm, we will first explain the *sequential formulation*. First, choose an arbitrary node in the graph as a root node. Then, starting in all leaf nodes, compute the *messages* μ, given by [7]

$$\mu_{a\to i}(x_i) = \sum_{x_{\mathsf{N}(a)\setminus\{i\}}} f_a(x_{\mathsf{N}(i)}) \prod_{j\in\mathsf{N}(a)\setminus\{i\}} \mu_{j\to a}(x_j) \tag{2.18}$$

$$\mu_{i\to a}(x_i) = \prod_{b\in\mathsf{N}(i)\setminus\{a\}} \mu_{b\to i}(x_i) \tag{2.19}$$

where the notation $\mathsf{N}(i)$ denotes the set of all neighbors of node i, and where i and j refer to variable indices and a and b refer to factor indices, such that $\mu_{a\to i}$ denotes the message from factor f_a to variable x_i and $\mu_{i\to a}$ denotes the message in the opposite direction. See Figure 2.2 for figures showing how the messages are passed along the edges of the factor graph. Each node passes a message onto its parent towards the root node based on the messages it receives from its children. In the example in Chapter 2.1.1 the temporary factor $\tilde{f}_2$ that we computed played exactly this role. When all messages have propagated upwards to the root node, perform the same procedure of computing messages, only in the opposite direction until all leaf nodes have received a message. The marginals $p(x_i)$



***(a)** Message passing from a variable x_i to the factor f_a.*

***(b)** Message passing from a factor f_a to the variable x_i.*

***Figure 2.2:** Figures comparing the two message definitions what are used when doing belief propagation on a factor graph.*

are then given as

$$p(x_i) \propto \prod_{a \in \mathsf{N}(i)} \mu_{a \to i}(x_i). \tag{2.20}$$

A final remark is that it can be shown that we can rescale the messages by any arbitrary, positive constant, and still arrive at the correct marginals [13]. This will be useful later.

2.3.2 The parallel formulation

The parallel formulation proceeds almost the same, except we send all messages between all nodes *concurrently*. First, we initialize all messages as

$$\mu_{i \to a}(x_i) = 1, \tag{2.21}$$

$$\mu_{a \to i}(x_i) = 1. \tag{2.22}$$

Then, for each iteration, update the messages according to

$$\mu_{a \to i}(x_i) \leftarrow \sum_{x_{\mathsf{N}(a) \setminus \{i\}}} f_a(x_{\mathsf{N}(a)}) \prod_{j \in \mathsf{N}(a) \setminus \{i\}} \mu_{j \to a}(x_j), \tag{2.23}$$

$$\mu_{i \to a}(x_i) \leftarrow \prod_{b \in \mathsf{N}(i) \setminus \{a\}} \mu_{b \to i}(x_i) \tag{2.24}$$

until they converge. For trees the messages are guaranteed to converge to the exact marginals after a number of iterations equal to the *diameter of the tree*, which is simply equal to the longest distance between two nodes in the tree [26].

Note that the argument about rescaling of messages also holds for the parallel formulation.

2.4 Loopy belief propagation

Many real-world problems are modelled with a probability distribution that contains cycles, or *loops*, and therefore does not satisfy the tree structure constraint of BP. However, considering the message definitions (2.23) and (2.24) in the parallel formulation are *local*, i.e. they only concern themselves with the neighbors of the current node and not the global structure of the graph, it is possible to do BP on such a "loopy" graph. This possibility was even suggested by Pearl himself [4]. The resulting algorithm is aptly called *loopy belief propagation*. Spectacularly, this has in many instances, e.g. in target tracking [8], [10]–[13], [17]–[21], worked surprisingly well at approximating the exact solution, and one of the earliest successful uses of it is in the context of error-correcting *turbo codes* [27].

2.4.1 LBP's relation to variational inference - The Bethe free energy

Initially, little was understood about properties of LBP such as the accuracy of the estimates it computes and if and when the algorithm even convergence. Perhaps the largest contribution to understanding the behavior of LBP came with the seminal work by Yedidia et. al [28], [29]. In order to elucidate the properties of LBP, its fixed points and its connection to *variational inference*, which is the theory of approximating *probability densities* by optimization, the following will be a condensed summary of their derivation.

The key first step is to turn to *statistical mechanics* by rewriting our loopy distribution $p(x_{\mathcal{V}}) \propto f(x_{\mathcal{V}})$ in terms of *Boltzmann's law*, which describes the state of a given system,

$$p(x_{\mathcal{V}}) = \frac{1}{Z} e^{-E(x_{\mathcal{V}})} \tag{2.25}$$

where Z denotes the normalization constant, also called the *partition function*, of our probability distributino $p(z_{\mathcal{V}})$ and

$$E(x_{\mathcal{V}}) = -\sum_{\nu} f_{\nu}(x_{\nu}) \tag{2.26}$$

denotes the energy of the system in state $x_{\mathcal{V}}$. A particular quantity of interest in statistical mechanics is the *Helmholtz free energy* F_H given by

$$F_H = -\ln Z \tag{2.27}$$

which is in general infeasible to compute exactly, and so much effort has been invested into approximating it. In particular, F_H can be approximated by Here Yedidia et. al proposes the use of the *variational free energy*, sometimes also called *Gibbs free energy*, defined for a *trial density* q as

$$F(q) = U(q) - H(q), \tag{2.28}$$

$$U(q) = \mathbb{E}_q[E(x_{\mathcal{V}})] \tag{2.29}$$

$$H(q) = -\mathbb{E}_q[\ln q(x_{\mathcal{V}})] \tag{2.30}$$

where $\mathbb{E}_q[\bullet]$ denotes the expectation under the density q, $U(q)$ denotes the *variational* *average energy* and $H(q)$ the *variational entropy*. It is possible to rewrite (2.28) as

$$F(q) = F_H + D(q||p) \tag{2.31}$$

where

$$D(q||p) = \mathbb{E}_q\left[\ln\frac{q(x_\mathcal{V})}{p(x_\mathcal{V})}\right] \tag{2.32}$$

is the *Kullback-Leibler divergence* between $q(x_\mathcal{V})$ and $p(x_\mathcal{V})$. As $D(q||p) \geq 0$, with $D(q||p)$ iff $q = p$, and F_H being constant, minimizing $F(q)$ will minimize $D(q||p)$ towards 0, yielding the q that best approximates p. This motivates the use of $F(q)$ as a cost function to minimize.

The last step remaining for calculating our trial distribution $q(x_\mathcal{V})$ is to construct it to take the form of a simpler function we can do inference on. The particular constraint we will choose is the $q(x_\mathcal{V})$ that factorizes as

$$q_{\text{Bethe}}(x_\mathcal{V}) = \frac{\prod_a q_a(x_{\mathsf{N}(a)})}{\prod_{i\in\mathcal{V}} q_i(x_i)^{d_i-1}} \tag{2.33}$$

where $\prod_a$ denotes the product over the same subsets of variables that makes up the factorization of our original distribution p and d_i denotes the $degree$ of node i, i.e. the number of adjacent edges. The choice of factorization in (2.33) is called the *Bethe approximation* [30]. This choice might seem arbitrary, but this factorization is exact for distributions that factorizes into trees [29]. Hence, in some sense, when optimizing we retrieve the function q that is the closest to a "tree" with the same nodes and edges as in our original distribution p. Note that in general q_{Bethe} will not be a proper distribution in the sense that its factorization as a tree over a graph with loops in general will be invalid.

All the pieces are now in place for optimization. We start by inserting our Bethe approximation q_{Bethe} into the variational free energy functional in (2.28) to get the *Bethe free energy* F_{Bethe}. We require that the factors q_a and q_i in (2.33) behave like distributions in the sense that they are nonnegative, sum to 1, and that they marginalize properly. To achieve this we turn to the theory of *constrained optimization* [31] by defining the *Lagrangian* $\mathcal{L}$ as

$$\begin{aligned}\mathcal{L} = F_{\text{Bethe}} &+ \sum_a \lambda_a\left[\sum_{x_\nu} q_\nu(x_\nu) - 1\right] + \sum_i \lambda_i\left[\sum_{x_i} q_i(x_\nu) - 1\right] \\ &+ \sum_{i'}\sum_{a\in\mathsf{N}(i')}\sum_{x_{i'}}\lambda_{a\to i'}\left[q_{i'}(x_{i'}) - \sum_{x_{\mathsf{N}(a)}\setminus\{i'\}} q_a(x_{\mathsf{N}(a)})\right]\end{aligned} \tag{2.34}$$

where $i' = \{i \in \mathcal{V} \mid d_i \geq 2\}$, i.e. the indices over all variable nodes with degree $d_i \geq 2$, λ_a, λ_i and $\lambda_{a \to i'}$ are *Lagrangian multipliers* and we have ignored the nonnegativity constraints as they are *inactive* during optimization and thus does not influence the solution [29]. The final insight is to solve for the stationary points of (2.34) and insert the definition

$$\lambda_{a \to i'} = \ln \prod_{b \in \mathsf{N}(i') \setminus \{a\}} \mu_{b \to i'}(x_{i'}) \tag{2.35}$$

such that, after the dust settles, we indeed arrive at the fixed-point equation (2.20) which is identical to the one in BP. This proves that when LBP converges to a fixed point, *the estimated marginals are a local minima of the Bethe free energy*.

2.4.2 Research results on properties of loopy belief propagation

Research into LBP is an active research field. A seminal paper with empirical tests and analysis of LBP is by Murphy et. al in [26]. Of particular interest is their investigation into divergence of LBP, and in their results small priors and weights in the graph are suspected as being the cause. In [32] by Ihler et. al they analyze accumulation of errors during LBP and relates this to the *dynamic range*, i.e. the largest ratio of the factors, of the graph, and derives convergence conditions for LBP.

2.4.3 Determining convergence of loopy belief propagation

Assuming that LBP will converge on a graph, we need a way of measuring convergence. A commonly used method for determining convergence, e.g. in [26], is by using the *max norm* between messages at different iterations, i.e., when the largest error between messages at different iterations reaches below some threshold, we terminate. In [13] they use a specialized distance metric for determining convergence based on the work in [32]. In the following we will use the metric in [13] since the LBP methods are based on their work.

3 | Data association in multitarget tracking

Traditionally, doing data association in MTT involves finding the possible association hypotheses, deriving a *joint association hypothesis distribution* and marginalizing this distribution using the found hypotheses. To be able to do this from a practical perspective, however, multiple model assumptions and simplifications are necessary. The following chapter will first introduce model assumptions that are made in MTT in order to do data association, and then derive the joint association hypothesis distributions that are used in the MTT algorithms *Joint probabilistic data association* (JPDA) and *Multiple hypothesis tracker* (MHT).

3.1 Multitarget tracking models and assumptions for data association

This section will first introduce the model assumptions that are necessary for track state estimation, which includes necessary expressions for deriving the joint association hypothesis distribution. Then, the more specific MTT model assumptions are introduced.

3.1.1 Track state estimation

We will quickly see when deriving the joint association hypothesis posterior in Chapter 3.2 that specifying the distributions for the track state $\boldsymbol{x}_k^t$ and measurement $\boldsymbol{z}_k$ is necessary. For the sake of simplicity we will therefore make *linear Kalman filter* assumptions and the resulting tools for data association will be summarized below. A full derivation and justification of this estimation framework is outside the scope of this text, and the reader is instead referred to [6], [33]–[36] for a more thorough treatment.

We use the notation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for a multivariate Gaussian distribution, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the parameters of the distribution, specifically the expectation value vector and covariance matrix, respectively. Due to the Kalman filter assumptions, we assume that the state of each track t evolve from timestep $k-1$ to k according to the *process model* $p(\boldsymbol{x}_k^t|\boldsymbol{x}_{k-1}^t)$ which we assume is linear and Gaussian, such that we have

$$\boldsymbol{x}_k^t = \mathbf{F}\boldsymbol{x}_{k-1}^t + \boldsymbol{v}_{k-1}, \tag{3.1a}$$

$$\boldsymbol{v}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{3.1b}$$

$$p(\boldsymbol{x}_k^t|\boldsymbol{x}_{k-1}^t) = \mathcal{N}\left(\mathbf{F}\boldsymbol{x}_{k-1}^t, \mathbf{Q}\right). \tag{3.1c}$$

The *measurement model* $p(\boldsymbol{z}_k|\boldsymbol{x}_k^t)$ is similarly given as

$$\boldsymbol{z}_k = \mathbf{H}\boldsymbol{x}_k^t + \boldsymbol{w}_k, \tag{3.2a}$$

$$\boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{3.2b}$$

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k^t) = \mathcal{N}\left(\mathbf{H}\boldsymbol{x}_k^t, \mathbf{R}\right). \tag{3.2c}$$

Assume that the *posterior distribution* $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ of a track t in timestep $k-1$, where $\boldsymbol{z}_{1:k-1} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{k-1}\}$ denotes all measurements from timestep 1 to $k-1$, is Gaussian. Together with the equations in (3.1) and (3.2), the *prior distribution* $p(\boldsymbol{x}_k^t|\boldsymbol{z}_{1:k-1})$ is given by

$$p(\boldsymbol{x}_k^t|\boldsymbol{z}_{1:k-1}) = \mathcal{N}\left(\hat{\boldsymbol{x}}_{k|k-1}^t, \mathbf{P}_{k|k-1}^t\right), \tag{3.3}$$

$$\hat{\boldsymbol{x}}_{k|k-1}^t = \mathbf{F}\hat{\boldsymbol{x}}_{k-1}, \tag{3.4}$$

$$\mathbf{P}_{k|k-1}^t = \mathbf{F}\mathbf{P}_{k-1}^t\mathbf{F}^\mathsf{T} + \mathbf{Q}, \tag{3.5}$$

the *likelihood distribution* $p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})$ is given by

$$p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}) = \mathcal{N}(\hat{\boldsymbol{z}}_k, \mathbf{S}_k), \tag{3.6}$$

$$\hat{\boldsymbol{z}}_k = \mathbf{H}\hat{\boldsymbol{x}}_{k|k-1}^t, \tag{3.7}$$

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}^t\mathbf{H} + \mathbf{R}, \tag{3.8}$$

and the new posterior distribution $p(\boldsymbol{x}_k^t|\boldsymbol{z}_{1:k})$ in timestep k is given by

$$\begin{aligned}
p(\boldsymbol{x}_k^t|\boldsymbol{z}_{1:k}) &= \mathcal{N}\left(\hat{\boldsymbol{x}}_k^t, \mathbf{P}_k^t\right), && (3.9)\\
\hat{\boldsymbol{x}}_k^t &= \hat{\boldsymbol{x}}_{k|k-1}^t + \mathbf{W}_k^t\left(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k\right) && (3.10)\\
\mathbf{P}_k^t &= \left(\mathbf{I} - \mathbf{W}_k^t\mathbf{H}\mathbf{P}_{k|k-1}^t\right)\mathbf{P}_{k|k-1}^t\left(\mathbf{I} - \mathbf{W}_k^t\mathbf{H}\mathbf{P}_{k|k-1}^t\right)^{\mathsf{T}}\\
&\quad + \mathbf{W}_k^t\mathbf{R}\left(\mathbf{W}_k^t\right)^{\mathsf{T}} && (3.11)\\
\mathbf{W}_k^t &= \mathbf{P}_{k|k-1}^t\mathbf{H}^{\mathsf{T}}\mathbf{S}_k^{-1}. && (3.12)
\end{aligned}$$

3.1.2 Overview of MTT specific models and concepts

The following will review the standard MTT modelling assumptions and concepts [14], [33], [37] which will be used when deriving the joint association hypothesis posteriors.

Multiple measurements

The main difference in MTT when estimating track state contrary to other state estimation cases, e.g. the assumptions made in the Kalman filter, is the fact that we receive *multiple* measurements each timestep, and has to infer the source of these measurements. The normal Kalman filter assumption is that we receive a single measurement $\boldsymbol{z}_k$ that we know for certain originates from the track we estimate, and that these arrive at a regular rate. In MTT we can not make the same assumptions due to the nature of the sensors used, as these are *exteroceptive*, meaning they observe some part of the surveillance region where tracks are allowed to exist.

Modifying the Kalman filter equations for this purpose requires minimal intervention. We use $\boldsymbol{z}_k^j$, $j \in \{1, \dots, m_k\}$ to denote the jth measurement out of the m_k we receive in timestep k. For likelihoods we are required to condition on an association between measurement j and track t, and we denote the corresponding predicted measurement and likelihood covariance by $\hat{\boldsymbol{z}}_k^{tj}$ and $\mathbf{S}_k^{tj}$, respectively. When conditioning on measurements we will instead condition on *measurement sets* $Z_{1:k} = \{Z_1, \dots, Z_k\}$, where the measurement set Z_l in timestep l is given as $Z_l = \{\boldsymbol{z}_k^1, \dots, \boldsymbol{z}_l^{m_l}\}$, for all timesteps.

The definition of a track

Before proceeding, it will be useful in the sequel to properly define the notion of a *track* and how it differs from a *target*. We will adhere to the convention used in [33]. A target will refer to an actual object in the surveillance region. A track will refer to a sequence of measurements or misdetections over time. More mathematically, assume we have k consecutive sets of measurements denoted by $Z_1 = \{\boldsymbol{z}_1^1, \dots, \boldsymbol{z}_1^{m_1}\}, \dots, Z_k = \{\boldsymbol{z}_k^1, \dots, \boldsymbol{z}_k^{m_k}\}$. A track t can then be represented as a vector

$$\mathcal{I}^t = [i_1, \dots, i_k] \tag{3.13}$$

where $i_l = \{0, \dots, m_l, N\}$ for each $l \in \{1, \dots, k\}$ [33, Definition 9.1.1, slightly modified], where we have added the *nonexistence index* N to indicate a track that has not been detected yet, and as such "does not exist". An important consequence of discriminating between a target and track this way is that a target remains semantically exactly one object in the real-world, while we can have multiple tracks for the same target. One can think of a track as a possible trajectory of a target given the measurements we have, and that there are multiple way of interpreting the measurements we have, hence multiple tracks.

Misdetections

Due to reasons such as the track being obscured by something or not in the field of view of the sensors, we occasionally get *misdetections*, i.e. that no measurement in the set of measurements received in a timestep originates from a given track. We will here model the detection of a track t as a *Bernoulli* distributed variable τ^t, with $\tau^t = 1$ defined as detection and $\tau^t = 0$ defined as misdetection, such that

$$\mathsf{Pr}\{\tau^t\} = \begin{cases} P_d, & \tau^t = 1, \\ 1 - P_d, & \tau^t = 0, \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

for some constant detection probability P_d.

Clutter

The second MTT modelling necessary is to model the measurements that remain unassociated to any tracks, called *clutter*. It is assumed that any excess measurements not from tracks are due to*false alarms*. Assuming that the field of view of the sensor can be divided into M cells with a constant and independent probability for false alarm equal to P_{FA} in each cell, the *clutter cardinality*, i.e. the number of clutter measurements φ_k, follows a *Binomial* distribution, such that

$$\mathsf{Pr}\{\varphi_k\} = \binom{M}{\varphi_k} P_{FA}^{\varphi_k} \left(1 - P_{FA}\right)^{M - \varphi_k}. \tag{3.15}$$

Taking the limits $M \to \infty$ and $P_{FA} \to 0$ while keeping $MP_{FA} = \lambda V_k$, it can be shown that the distribution in (3.15) becomes a *Poisson* distribution with parameter $\Lambda = \lambda V_k$,

$$\mathsf{Pr}\{\varphi_k\} = e^{-\lambda V_k} \frac{(\lambda V_k)^{\varphi_k}}{\varphi_k!} \tag{3.16}$$

where we call λ the *clutter intensity* and V_k is the volume of the measurement space where measurements can exist. Hence, by assuming that the sensor in use has sufficiently high resolution and sufficiently low false alarm rate, the Poisson approximation is valid, and is the modelling assumption that will be made going forward. Making this approximation has its weaknesses, such as having a "flatter" distribution than the Binomial distribution, but the modelling simplifications from using a Poisson rather than a Binomial distribution justifies it. Additionally, an important benefit of using the Poisson distribution for modelling clutter is that the number of clutter in one measurement region is independent of all other regions, which is a natural assumption for clutter.

Specifically choosing a homogeneous Poisson distribution, i.e. a Poisson distribution with constant intensity λ, has another added benefit. In this case, given how many clutter measurements there are, the clutter is *uniformly* distributed in the measurement space. In other words, the measurement model is given simply as

$$p(\boldsymbol{z}_k^j|\boldsymbol{x}_k^t) = p(\boldsymbol{z}_k^j) = \frac{1}{V_k}, \tag{3.17}$$

where we have used the assumption that all clutter measurements are independent of track state.

Undetected targets

An obvious, but important, distinction between STT and MTT is that in MTT we allow for new targets to enter our surveillance region, which needs to be modelled. Strictly speaking, we need to distinguish between *newly arriving targets* and *undetected targets*. The distinction comes from the observation that targets that arrive in the surveillance region are not necessarily detected immediately. Namely, we can only model the number of *undetected targets in the surveillance region that we detect* β_k. We will use a simplified version of the model used in [14] which is a Poisson distribution given by

$$\mathsf{Pr}\{\beta_k\} = e^{-P_d \nu_k V_k} \frac{(P_d \nu_k V_k)^{\beta_k}}{\beta_k!} \tag{3.18}$$

where P_d is the detection probability from (3.14) and ν_k denotes the arrival intensity of new targets in all of the valid target space, which varies with time. The scaling by P_d is done to account for the fact that new targets will seemingly "arrive less often" than ν_k as we depend on detecting them with our sensor.

Extended object tracking and at-most-one assumption

The last, grossly simplifying assumptions that will be made are that *each target at most generates one measurement* and that *each measurement at most originates from one target*. It is easy to imagine scenarios where this obviously does not hold, for instance when a large vessel is passing by our sensor and the sensor returns a point cloud of detections, or that two small vessels are sufficiently close enough to overlap and return a single, merged detection. Making the assumptions, however, severely reduces the hypothesis space, which amongst other things allows for significantly more efficient marginalization. A tracker that integrates the fact that multiple measurement may originate from the same target uses *extended object tracking*, and is an open-research field. The topic is not explored further in this report and the reader is instead referred to [38].

3.1.3 Gating of measurements

In practice we will not consider all measurements for each track t, but instead all *gated* measurements $\boldsymbol{z}_k^j$, which indicates measurements that pass the χ^2-test

$$\varepsilon^{tj} < g^2 \tag{3.19}$$

where we call g the number of sigmas we consider and ε^{tj} denotes the *Normalized innovation squared* (NIS) and is given by the parameters of the likelihood distribution in (3.6) as

$$\varepsilon^{tj} = (\boldsymbol{z}_k^j - \hat{\boldsymbol{z}}_k^{tj})^\mathsf{T}(\mathbf{S}_k^{tj})^{-1}(\boldsymbol{z}_k^j - \hat{\boldsymbol{z}}_k^{tj}) \tag{3.20}$$

which indeed is χ^2-distributed as long as $\boldsymbol{z}_k^j$ is Gaussian with expectation $\hat{\boldsymbol{z}}_k^{tj}$ and covariance $\mathbf{S}_k^{tj}$. Note that in the following we will assume that all measurements are gated to simplify the notation and derivation.

The region of valid measurements, be it the entire measurement space or those that pass the χ^2-test in (3.19), is called the *validation gate*. In the case that we do gating of measurements, the volume V_k will refer to this, although we will see in the coming derivations that we will not be required to explicitly calculate it.

3.2 Hypothesis generation in MTT

Having derived the necessary tools and assumptions for data association, the next step is to derive the *association hypothesis marginals* π_k^{tj} that are used, which are the marginal probabilities that each track t should be associated to measurement j in timestep k. The reason is as follows. As the final step of the track states update in MTT, under the assumptions in Chapter 3.1, we compute the posterior for each track t as a *Gaussian mixture* over all track associations a_k^t, which, by total probability, is given by

$$p(\boldsymbol{x}_k^t|Z_{1:k}) = \sum_{j=0}^{m_k} \mathsf{Pr}\{a_k^t = j|Z_{1:k}\}\, p(\boldsymbol{x}_k^t|a_k^t = j, Z_{1:k}) \tag{3.21}$$

$$= \sum_{j=0}^{m_k} \pi_k^{tj} \mathcal{N}\left(\hat{\boldsymbol{x}}_k^{tj}, \mathbf{P}_k^{tj}\right) \tag{3.22}$$

where $\hat{\boldsymbol{x}}_k^{tj}$ and $\mathbf{P}_k^{tj}$ are the parameters of the association-conditioned posterior distribution of track t when the prior is updated with measurement j,

$$p(\boldsymbol{x}_k^t|a_k^t = j, Z_{1:k}) \propto p(\boldsymbol{x}_k^t|Z_{1:k-1})p(\boldsymbol{z}_k^j|\boldsymbol{x}_k^t), \tag{3.23}$$

where under linear Gaussian assumptions the prior $p(\boldsymbol{x}_k^t|Z_{1:k-1})$ and measurement model $p(\boldsymbol{z}_k^j|\boldsymbol{x}_k^t)$ are given by (3.3) and (3.2c), respectively, and π_k^{tj} is the association hypothesis posterior marginal for associating track t with measurement j,

$$\pi_k^{tj} = \mathsf{Pr}\{a_k^t = j \mid Z_{1:k}\} \tag{3.24}$$

$$= \sum_{a_k : a_k^t = j} \mathsf{Pr}\{a_k \mid Z_{1:k}\}\,, \tag{3.25}$$

where $a_k = \{a_k^1, \ldots, a_k^{n_k}\}$ is the set of all n_k track associations in timestep k and $a_k : a_k^t = j$ stands for all a_k where $a_k^t = j$. The measurement $j = 0$ indicates misdetection and in this case the posterior is equal to the prior distribution, i.e., $\hat{\boldsymbol{x}}_k^{tj} = \hat{\boldsymbol{x}}_{k|k-1}^t$ and $\mathbf{P}_k^{tj} = \mathbf{P}_{k|k-1}^t$.

Thus, our objective is to calculate the association hypothesis posterior marginals in (3.25) for each track t, $t \in \{1, \ldots, n_k\}$ and measurement j, $j \in \{1, \ldots, m_k\}$.

3.2.1 Hypothesis generation in JPDA

First described in [37] is the JPDA filter that builds upon the original *Probabilistic data association filter* (PDAF) [39] by extending the association marginals to accomodate interfering targets that gate the same measurements. We will here concern ourselves with how JPDA does the measurement updated of target states based on association hypothesis posterior, together with how it's computed.

Computation of joint association hypothesis posterior

Although the original derivation can be found in [37], the following will draw inspiration from [33]. The posterior is first calculated by Bayes' rule,

$$\begin{aligned}\mathsf{Pr}\{a_k \mid Z_{1:k}\} &= \mathsf{Pr}\{a_k \mid Z_k, m_k, Z_{1:k-1}\} && (3.26)\\ &\propto p(Z_k \mid a_k, m_k, Z_{1:k-1})\mathsf{Pr}\{a_k \mid m_k, Z_{1:k-1}\}\,, && (3.27)\end{aligned}$$

where the cardinality of Z_k, m_k, is used to make $p(Z_k \mid a_k, m_k, Z_{1:k-1})$ more tangible to work with. The first factor can be written as

$$p(Z_k \mid a_k, m_k, Z_{1:k-1}) = \int p(Z_k \mid a_k, m_k, \boldsymbol{x}_k^{1:n_k}) p(\boldsymbol{x}_k^{1:n_k} \mid a_k, m_k, Z_{1:k-1})\, \mathrm{d}\boldsymbol{x}_k^{1:n_k} \qquad (3.28)$$

by total probability, where $\boldsymbol{x}_k^{1:n_k} = \boldsymbol{x}_k^1, \boldsymbol{x}_k^2, \ldots, \boldsymbol{x}_k^{n_k}$ denotes the states of all tracks and we invoked the *Markov assumption* that is used in the Kalman filter to perform the substitution $p(Z_k \mid a_k, m_k, \boldsymbol{x}_k^{1:n_k}, Z_{1:k-1}) = p(Z_k \mid a_k, m_k, \boldsymbol{x}_k^{1:n_k})$. Conditioned on the association hypothesis a_k, we can partition the likelihoods based on the associations made for each measurement $\boldsymbol{z}_k^j$, assuming independence between them, such that

$$\begin{aligned}p(Z_k \mid a_k, m_k, Z_{1:k-1}) &= p(\boldsymbol{z}_k^1, \ldots, \boldsymbol{z}_k^{m_k} \mid a_k, m_k, Z_{1:k-1}) && (3.29)\\ &= \prod_{j=1}^{m_k} p(\boldsymbol{z}_k^j \mid a_k, Z_{1:k-1}). && (3.30)\end{aligned}$$

Inserting (3.30) into (3.28) and rearranging gives

$$\int p(Z_k \mid a_k, m_k, \boldsymbol{x}_k^{1:n}) p(\boldsymbol{x}_k^{1:n} \mid a_k, m_k, Z_{1:k-1}) \, \mathrm{d}\boldsymbol{x}_k^{1:n} \tag{3.31}$$

$$= \prod_{t:a_k^t=0} \left(\frac{1}{V_k} \underbrace{\int p(\boldsymbol{x}_k^t \mid a_k, Z_{1:k-1}) \, \mathrm{d}\boldsymbol{x}_k^t}_{=1} \right)$$

$$\cdot \prod_{t:a_k^t>0} \underbrace{\int p(\boldsymbol{z}_k^j \mid a_k, \boldsymbol{x}_k^t) p(\boldsymbol{x}_k^t \mid a_k, Z_{1:k-1}) \, \mathrm{d}\boldsymbol{x}_k^t}_{=l^{ta_k^t}} \tag{3.32}$$

$$= \frac{1}{V_k^{\varphi_k}} \cdot \prod_{t:a_k^t>0} l^{ta_k^t} \tag{3.33}$$

where $l^{ta_k^t}$ is the event-conditional likelihood for target t given association to measurement $a_k^t = j$, $t : a_k^t = 0$ denotes the set of all misdetected tracks t and $t : a_k^t > 0$ denotes the set of all detected tracks, V_k is the volume of the validation gate and we made the assumption that the track states are independent given the association hypothesis a_k, such that [33]

$$p(\boldsymbol{x}_k^{1:n_k} | a_k, Z_{1:k}) \propto \prod_{t:a_k^t=0} p(\boldsymbol{x}_k^t | Z_{1:k-1}) \prod_{t:a_k^t>0} p(\boldsymbol{z}_k^{a_k^t} | \boldsymbol{x}_k^t) p(\boldsymbol{x}_k^t | Z_{1:k-1}). \tag{3.34}$$

In order to construct the association prior $\mathsf{Pr}\{a_k \mid m_k\}$ we first introduce the detection variable τ_k such that

$$\tau_k^t = \begin{cases} 1 & \text{if target } t \text{ is detected in timestep } k \\ 0 & \text{if target } t \text{ is not detected in timestep } k \end{cases} \tag{3.35}$$

where the dependency on $Z_{1:k-1}$ is dropped as, given only m_k, we have no information on how previous measurements should affect our belief for how to associate measurements in the current timestep [33]. As the detection event τ_k is entirely contained in the association event a_k, we can then rewrite the association hypothesis prior as

$$\mathsf{Pr}\{a_k \mid m_k\} = \mathsf{Pr}\{a_k, \tau_k \mid m_k\} \tag{3.36}$$

$$= \mathsf{Pr}\{a_k \mid \tau_k, m_k\} \, \mathsf{Pr}\{\tau_k \mid m_k\} \tag{3.37}$$

$$\propto \mathsf{Pr}\{a_k \mid \tau_k, m_k\} \, \mathsf{Pr}\{m_k \mid \tau_k\} \, \mathsf{Pr}\{\tau_k\} \, . \tag{3.38}$$

Let $\delta_k = \sum_{t=1}^{n_k} \tau_k^t$ denote the number of detected tracks and $\bar{\delta}_k = n_k - \delta_k$ the number of undetected tracks. The prior probability $\mathsf{Pr}\{\tau_k\}$ is given by

$$\mathsf{Pr}\{\tau_k\} = (1 - P_d)^{\bar{\delta}_k} P_d^{\delta_k} \tag{3.39}$$

where the detection distribution in (3.14) is used and is simply the probability that the δ_k tracks indicated by τ_k are detected and the remaining are not. The probability $\mathsf{Pr}\{m_k \mid \tau_k\}$ can be found by using that $m_k = \varphi_k + \delta_k$, where δ_k is a constant given τ_k such that $m_k \mid \tau_k$ is just a linear transform of the Poisson distributed variable φ_k, which is just a new Poisson distributed variable. Hence, in this case, the probability distribution is given by

$$\begin{align}
\mathsf{Pr}\{m_k \mid \tau_k\} &= e^{-\lambda V_k} \frac{(\lambda V_k)^{m_k - \delta_k}}{(m_k - \delta_k)!} \tag{3.40} \\
&= e^{-\lambda V_k} \frac{(\lambda V_k)^{\varphi_k}}{\varphi_k!}. \tag{3.41}
\end{align}$$

Lastly, $\mathsf{Pr}\{a_k \mid \tau_k, m_k\}$ can be found as the uniform distribution over all valid association hypotheses given τ_k and m_k,

$$\mathsf{Pr}\{a_k \mid \tau_k, m_k\} = \frac{1}{|\mathcal{A}_k|} \tag{3.42}$$

where $\mathcal{A}_k$ is the set of all valid association hypotheses a_k in timestep k. As all undetected tracks are associated to misdetection, this can only be done one way, and so the number of association hypotheses reduces to the number of permutations of detected tracks to measurements. With m_k measurements and δ_k detections, this amounts to

$$\begin{align}
|\mathcal{A}_k| &= m_k \cdot (m_k - 1) \cdot \ \ldots \ \cdot (m_k - (\delta_k + 1)) \tag{3.43} \\
&= \frac{m_k!}{(m_k - \delta_k)!} \tag{3.44} \\
&= \frac{m_k!}{\varphi_k!} \tag{3.45}
\end{align}$$

different possible association hypotheses, where the fact that $m_k = \varphi_k + \delta_k$ is used. Thus, inserting (3.45) into (3.42) gives

$$\mathsf{Pr}\{a_k \mid \tau_k, m_k\} = \frac{\varphi_k!}{m_k!}. \tag{3.46}$$

In conclusion, the association hypothesis prior $\mathsf{Pr}\{a_k \mid m_k\}$ can be written as

$$\mathsf{Pr}\{a_k \mid m_k\} \propto \frac{\varphi_k!}{m_k!} \cdot e^{-\lambda V} \frac{(\lambda V)^{\varphi_k}}{\varphi_k!} \cdot (1 - P_d)^{\bar{\delta}_k} P_d^{\delta_k} \tag{3.47}$$

$$\propto (\lambda V)^{\varphi_k} \cdot (1 - P_d)^{\bar{\delta}_k} P_d^{\delta_k} \tag{3.48}$$

where $\varphi!$ cancels out and $1/m_k!$ and $e^{-\lambda V}$ are moved into the proportionality as they are constant with respect to a_k.

Taking the product of (3.48) and (3.33) and moving $P_d^{\delta_k}$ into the product $\prod_{t:a_k^t>0} l^{ta_k^t}$, as both are over all δ_k detected tracks,

$$P_d^{\delta_k} \prod_{t:a_k^t>0} l^{ta^t} = \prod_{t:a_k^t>0} P_d l^{ta^t}, \tag{3.49}$$

then gives

$$\mathsf{Pr}\{a_k \mid Z_{1:k}\} \propto \lambda^{\varphi_k} (1 - P_d)^{\bar{\delta}_k} \prod_{t:a_k>0} P_D l^{ta_k^t} \tag{3.50}$$

where the V^{φ_k} cancels out. The last step is to rewrite the expression in (3.50) in the following way. Given Z_k, m_k is constant and equal to $m_k = \delta_k + \varphi_k$, such that we can rewrite the clutter contribution factor λ^{φ_k} as

$$\lambda^{\varphi_k} = \lambda^{\varphi_k} \cdot \frac{\lambda^{m_k}}{\lambda^{\varphi_k + \delta_k}} \tag{3.51}$$

$$= \lambda^{\varphi_k} \cdot \frac{\lambda^{m_k}}{\lambda^{\varphi_k} \lambda^{\delta_k}} \tag{3.52}$$

$$\propto \frac{1}{\lambda^{\delta_k}} \tag{3.53}$$

where we use that λ^{m_k} is just a constant we can hide in the proportionality sign such that we can rewrite (3.50) as

$$\mathsf{Pr}\{a_k \mid Z_{1:k}\} \propto (1 - P_d)^{\bar{\delta}_k} \prod_{t:a_k>0} \frac{P_D l^{ta_k^t}}{\lambda}. \tag{3.54}$$

3.2.2 Hypothesis generation in MHT

MHT was first presented in [40] as a multiple-scan filter which, contrary to JPDA, does not reduce the multiple hypotheses for each posterior of each track into one posterior, but instead keeps track of the M best (i.e., most likely) hypotheses at all times and develops new hypotheses posteriors in a recursive manner.

To better understand how MHT structures association hypotheses as *trees*, we will consider as an example the multitarget tracking data association scenario in Figure 3.1. The root is some parent hypothesis from the previous timestep $\theta^{l}_{1:k-1}$, where we assume we have L parent hypotheses and $l \in \{1, \dots, L\}$. The different possible child hypotheses that can be formed based on the measurement set Z_k are then formed as branches, where the depth of the tree is the number of measurements and each node on the layer describes the origin of the current measurement. In the original formulation of Reid, the valid origins of a measurement can be either clutter, denoted by $j = 0$, a track not associated yet and where the measurement passes the gating criterion given in Chapter 3.1.3, denoted by $j \in \{1, \dots, n_t\}$, or a new target, denoted by $j \in \{n_t + 1, \dots, n_t + m_k\}$, where n_t is number of existing tracks.

Defining hypotheses as sets of tracks

Assume we get in total R child hypotheses and denote some arbitrary child hypothesis by $\theta^{r}_{1:k}$ with $r \in \{1, \dots, R\}$. The association hypothesis a_k refers to the associations we make between tracks and measurements in a branch from $\theta^{l}_{1:k-1}$ to $\theta^{r}_{1:k}$ in the hypothesis tree and where the parent index l and child index r are clear from the context. A hypothesis needs to contain the full information of all associations made between tracks and measurements for all timesteps k, and so we will use the recursive definition

$$\theta^{r}_{1:k} = a_k \cup \theta^{l}_{1:k-1} \tag{3.55}$$

with base case $\theta^{1}_{0} = \{\ \}$. Since we define tracks as a collection of measurement associations or misdetections as in Chapter 3.1.2, we can refer to a hypothesis as a subset of all n_k track indices

$$\theta^{r}_{1:k} \subseteq \{1, \dots, n_k\} \tag{3.56}$$

where instead each index $t \in \theta^{r}_{1:k}$ points to a vector $\mathcal{I}^t$ over measurement associations or misdetections for track t as defined in (3.13).

More informally, the hypothesis can be defined as “the set of all tracks that have been initialized in the hypothesis at some point up the branch in the full hypothesis tree”. When a track is contained in the hypothesis, we will say that the track *exists* in the hypothesis. Conversely, this implies that *nonexistence* means the track is not contained in the hypothesis. From the definition (3.56) we will allow the notations $t \in \theta$ to indicate tracks t that exist in the hypothesis θ and $t \notin \theta$ to indicate tracks t that does not exist in the hypothesis θ.

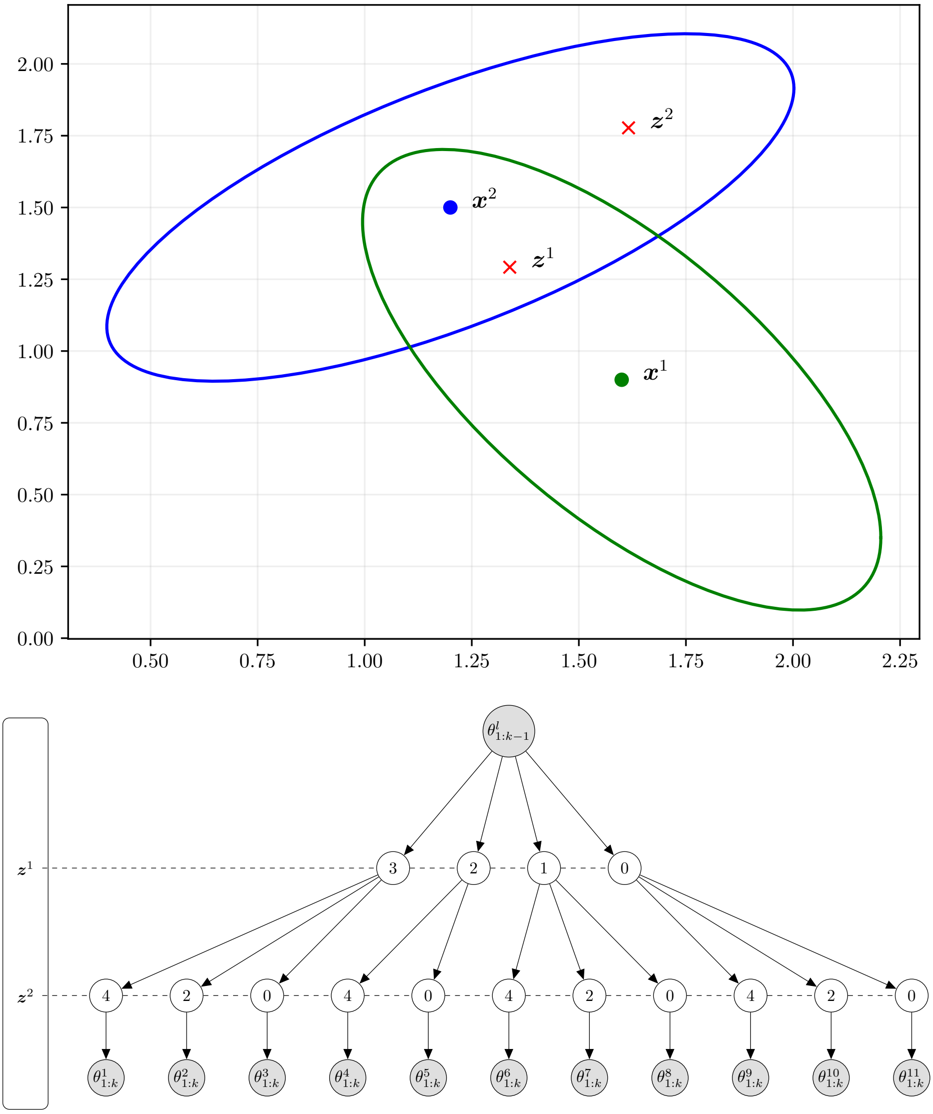


Figure 3.1: *Example of a simple association problem in MTT and how a measurement-oriented hypothesis tree is constructed from a parent hypothesis $\theta^l_{1:k-1}$ into eleven new child hypotheses $\theta^{1:11}_{1:k}$ in the same manner as in [40]. Each layer in the tree corresponds to different associations to the same measurement, indicated by the dashed lines, such that the layer index together with the index in the node corresponds to an association. The index 0 is used to indicate misdetection while the indices 3 and 4 refer to the new track index that is initialized in the unassociated measurement. A full hypothesis can be retrieved by traversing the tree from the root to a leaf, where each branch is a hypothesis. Note also that we chose to enumerate the child hypotheses by $1, 2, \ldots, 11$ for convenience, but that it of course is not necessary in general.*

Computing the joint association hypothesis posterior

Following the tree structure, the association hypothesis joint posteriors for each child hypothesis are found by computing the probability for each leaf node hypothesis for each possible parent root hypothesis. By writing $\theta^r_{1:k}$ as the union of the parent hypothesis $\theta^l_{1:k-1}$ and the association hypothesis a_k the distribution $\mathsf{Pr}\{\theta^r_{1:k} \mid Z_{1:k}\}$ can be written out as

$$\begin{aligned}\mathsf{Pr}\{\theta^r_{1:k} \mid Z_{1:k}\} &= \mathsf{Pr}\big\{a_k, \theta^l_{1:k-1} \mid Z_{1:k-1}, Z_k\big\} && (3.57)\\ &\propto p(Z_k \mid a_k, \theta^l_{1:k-1}, Z_{1:k-1})\mathsf{Pr}\big\{a_k \mid \theta^l_{1:k-1}, Z_{1:k-1}\big\}\\ &\qquad \mathsf{Pr}\big\{\theta^l_{1:k-1} \mid Z_{1:k-1}\big\}\,. && (3.58)\end{aligned}$$

The last factor in (3.58) sets up the recursion, and so we will only concern ourselves with the two first factors. The likelihood $p(Z_k \mid a_k, \theta^l_{1:k-1}, Z_{1:k-1})$ is, given the child hypothesis a_k, just a product over all associated and unassociated likelihoods such that

$$\begin{aligned}p(Z_k \mid a_k, \theta^l_{1:k-1}, Z_{1:k-1}) &= \prod_{j=1}^{m_k} p(\boldsymbol{z}^j_k \mid a_k, \theta^l_{1:k-1}, Z_{1:k-1}) && (3.59)\\ &= \frac{1}{V_k^{\varphi_k}} \cdot \prod_{t:a^t>0} l^{ta^t}, && (3.60)\end{aligned}$$

similarly to (3.33) in JPDA. The prior distribution for the child hypothesis a_k is slightly more involved to derive. As the only information we have on the measurement set Z_k is the cardinality m_k, all possible hypotheses are equally likely, and so the derivation reduces to arguments using combinatorics counting the possible hypotheses. The hypothesis event can be partitioned into what Reid calls a *number event*, *configuration event* and *association event.*

The number event is computing the probability of receiving $m_k = \delta_k + \varphi_k + \beta_k$ measurements, where δ_k, φ_k and β_k are number of detections of known tracks, number of clutter measurements and number of new targets detected, respectively, and where φ_k and β_k are both Poisson distributed as in (3.16) and (3.18) and each detection is a Bernoulli variable as in (3.14). Denoting number of known tracks by N_{TGT}, which is a constant given the parent hypothesis $\theta^l_{1:k-1}$, the number event distribution is then

$$\mathsf{Pr}\big\{\delta_k, \varphi_k, \beta_k \mid \theta^l_{1:k-1}\big\} = \binom{N_{TGT}}{\delta} P_d^{\delta}(1-P_d)^{N_{TGT}-\delta_k} \frac{(P_d\nu_k V_k)^{\beta_k}}{\beta_k!} e^{-P_d\nu_k V_k} \frac{(\lambda V_k)^{\varphi_k}}{\varphi_k!} e^{-\lambda V_k}. \tag{3.61}$$

The configuration event $\mathcal{C}_k$ is the uniform distribution over all different ways that the

m_k measurements can categorized as either detected track, clutter or new target. As this is given as

$$\binom{m_k}{\delta_k}\binom{m_k-\delta_k}{\varphi_k}\binom{m_k-\delta_k-\varphi_k}{\beta_k}=\binom{m_k}{\delta_k}\binom{m_k-\delta_k}{\varphi_k} \tag{3.62}$$

different ways, where $\binom{m_k-\delta_k-\varphi_k}{\beta_k}=\binom{\beta_k}{\beta_k}=1$ is used, the configuration event distribution is given as

$$\mathsf{Pr}\{\mathcal{C}_k \mid \delta_k,\varphi_k,\beta_k\}=\frac{1}{\binom{m_k}{\delta_k}\binom{m_k-\delta_k}{\varphi_k}}=\frac{\delta_k!\beta_k!\varphi_k!}{m_k!}. \tag{3.63}$$

Lastly, as each track is unique, the association event α_k counts the number of permutations the detected tracks can be associated to the established tracks, which gives the uniform probability

$$\mathsf{Pr}\{\alpha_k \mid \mathcal{C}_k\}=\left[\frac{N_{TGT}!}{(N_{TGT}-\delta_k)!}\right]^{-1}=\frac{(N_{TGT}-\delta_k)!}{N_{TGT}!}. \tag{3.64}$$

Combining (3.60), (3.61), (3.63) and (3.64), canceling factors and moving constants into the proportionality then gives the joint association hypothesis posterior

$$\mathsf{Pr}\{\theta^r_{1:k}|Z_{1:k}\}\propto\mathsf{Pr}\{\theta^l_{1:k-1}|Z_{1:k-1}\}\,P_d^{\delta_k}\lambda^{\varphi_k}(P_d\nu_k)^{\beta_k}(1-P_d)^{N_{TGT}-\delta_k}\prod_{t:a^t_k>0}l^{ta^t}. \tag{3.65}$$

Reducing the hypothesis space by introducing track existence

The association scenario in Figure 3.1 is very simple, but still the hypothesis tree branches out quickly. The association tree only continues doing this exponentially for more common and more complex association scenarios in MTT, and so any means of reducing the hypothesis branching is of interest. In more modern tracking methods such as *Integrated probabilistic data association* (IPDA) [41], its multi-target tracking counterpart *Joint integrated probabilistic data association* (JIPDA) [42] and PMBM [14], the notion of *track existence* is introduced, and in particular in PMBM this is used to make the hypothesis space more compact. A full explanation of the above-mentioned filters is outside the scope of this text, and so the following will only explain how hypothesis enumeration is done in PMBM, as it is also a multihypothesis tracker.

As seen above, in MHT we explicitly distinguish between a measurement being declared clutter or a new target. What can be seen is that except for this difference, *the two hypotheses containing these declarations of measurement origin will otherwise* *always be identical*. PMBM does the clever trick of simply combining the two hypotheses into one hypothesis by combining the number of new targets detected β_k and clutter φ_k into one variable $\kappa_k = \beta_k + \varphi_k$. The idea to combine β_k and φ_k was first proposed by Bar-Shalom et. al in [43]. Instead, PMBM always initializes a new track together with an extra state called the *track existence probability* r_t. A full explanation for how this track existence probability is initialized and used in the filter is outside the scope of this text, and the reader is instead referred to [44]. We will, however, to be consistent with the true hypothesis distribution in PMBM, modify the detection probability P_d in the misdetection probability and detection likelihood for a track to be $r_t P_d$, i.e. the product of the original detection probability and the track existence probability.

Based on the discussion above, by combining these two declarations into one gives the hypothesis tree in Figure 3.2, where the number of leaf nodes is significantly reduced. Considering that both the number of new targets and clutter are Poisson distributed and

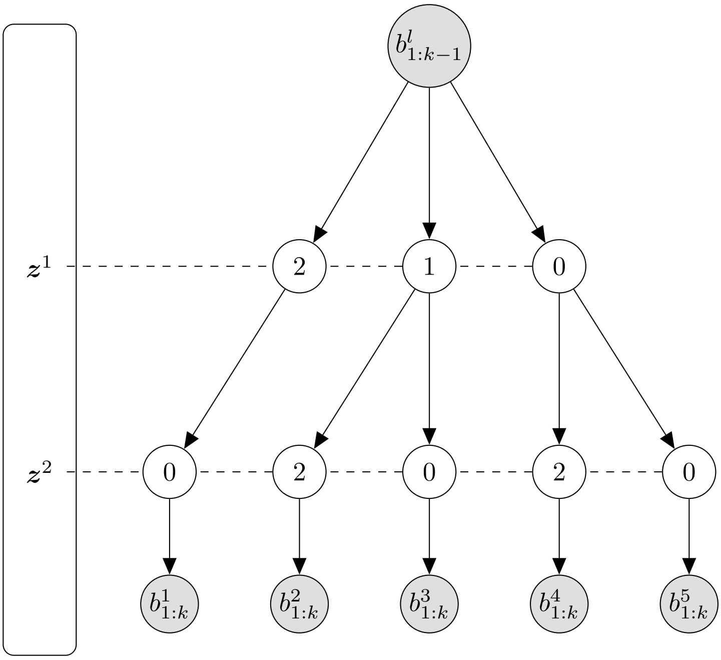


Figure 3.2: *Hypothesis tree based on the scenario in Figure 3.1 but where we only enumerate new targets. Comparing the number of leaf nodes in the two examples, the benefits of using the combined clutter and new target declaration is evident, as this restricts the growth of the hypothesis tree.*

independent, we can combine the two variables into one by summing them together, creating a new Poisson distributed variable $\kappa_k = \beta_k + \varphi_k$ with distribution

$$\mathsf{Pr}\{\kappa\} = e^{-(P_d\nu+\lambda)V_k} \frac{\left[(P_d\nu + \lambda)\, V_k\right]^{\kappa_k}}{\kappa_k!} \tag{3.66}$$

where P_d is the detection probability and ν_k is the arrival rate of new targets as defined in (3.18), λ is the clutter rate as defined in (3.16) and V_k is the volume of the measurement region we consider. The joint association hypothesis posterior in PMBM can be shown to be almost on the same form as in (3.65), but with the combined new target and clutter it instead takes the form

$$\mathsf{Pr}\{\theta^r_{1:k}|Z_{1:k}\} \propto \mathsf{Pr}\left\{\theta^l_{1:k-1}|Z_{1:k-1}\right\} (r_t P_d)^{\delta_k} (P_d\nu{+}\lambda)^{\kappa_k} (1{-}r_t P_d)^{N_{TGT}-\delta_k} \prod_{t:a^t_k>0} l^{ta^t}. \tag{3.67}$$

It is, however, common to rewrite (3.67) in the following way, similarly to how we did for JPDA in (3.64). First, we combine the product $(r_t P_d)^{\delta_k}$ and $\prod_{t:a^t_k>0} l^{ta^t}$, as both are products over all detected tracks. Additionally, we have that the number of measurements m_k is constant when conditioning on the measurement set Z_k and $m_k = \kappa_k + \delta_k$, so

$$\begin{aligned}
(P_d\nu + \lambda)^{\kappa_k} &= (P_d\nu + \lambda)^{\kappa_k} \frac{(P_d\nu + \lambda)^{m_k}}{(P_d\nu + \lambda)^{\kappa_k+\delta}} && (3.68)\\
&= (P_d\nu + \lambda)^{\kappa_k} \frac{(P_d\nu + \lambda)^{m_k}}{(P_d\nu + \lambda)^{\kappa_k}(P_d\nu + \lambda)^{\delta_k}} && (3.69)\\
&\propto \frac{1}{(P_d\nu + \lambda)^{\delta_k}}, && (3.70)
\end{aligned}$$

where we use that $(P_d\nu + \lambda)^{m_k}$ is just a constant we can hide away in the proportionality sign. We therefore substitute $(P_d\nu + \lambda)^{\kappa_k}$ with (3.70) in (3.67), which allows us to combine it with (3.49). Lastly, by using the notation $\bar{\delta}_k = N_{TGT} - \delta_k$ for the number of undetected tracks we therefore arrive at the expression

$$\mathsf{Pr}\{\theta^r_{1:k}|Z_{1:k}\} \propto \mathsf{Pr}\left\{\theta^l_{1:k-1}|Z_{1:k-1}\right\} (1 - r_t P_d)^{\bar{\delta}_k} \prod_{t:a^t_k>0} \frac{r_t P_d l^{ta^t}}{P_d\nu + \lambda} \tag{3.71}$$

for the joint association hypothesis posterior in PMBM. For the remainder of the report we will use this form for marginalization.

3.3 Marginalization of joint association hypothesis posterior

After having derived the different forms of the joint association hypothesis posterior, what remains is to compute the desired association marginals. Namely, two methods will be described, the first being how to compute them exactly and an approximate solution that is more common in existing implementations of MTT trackers, *Murty's method* in conjunction with a linear assignment solver.

3.3.1 Explicit hypothesis enumeration

The most straight-forward way of computing the marginals is to enumerate all possible hypotheses by traversing all branches of the hypothesis tree and accumulate the marginals while traversing. Since each hypothesis tree, as they appear in Figure 3.2, is dependent on the prior hypothesis, i.e. the root node, these marginal computations have to be repeated for each prior hypothesis and combined with total probability. We will for the time being postpone how to compute exact marginals for the full multihypothesis problem and instead only present the algorithm for exact marginal computation in a single-hypothesis case.

Although not the most efficient way of doing hypothesis enumeration, we illustrate how this can be done in the algorithm ExactMarginalization which can be found in Algorithm 2 and uses two subalgorithms, HypothesisEnumeration in Algorithm 2 and TraverseHypothesisTree in Algorithm 3.

In Algorithm 3 we traverse the hypothesis tree as it is defined in MHT, which is *measurement-oriented*. This means that that we only explicitly enumerate what tracks a measurement j is associated with for each j, with $t = 0$ denoting clutter. Such measurement-oriented hypothesis is denoted by b, which corresponds to a branch in the hypothesis tree. All branches of the tree are then collected in the set $\mathcal{B}$. Note that in Algorithm 3 assumes that we are working on *references* to the actual variables $\mathcal{B}$ and b, such that modifying them in a procedure call modifies them in all prior calls in the call stack.

Since the joint association hypothesis posterior considers detections of tracks, we need to convert the set $\mathcal{B}$ to *track-oriented* hypotheses a, which are collected in the set $\mathcal{A}$. This happens in Algorithm 2.

Algorithm 1 Marginalization by explicit hypothesis enumeration when single-hypothesis.

```
procedure ExactMarginalization($Z_k$)
    $\mathbf{P} \leftarrow \mathbf{0}_{n_k \times (m_k+2)}$        ▷ Matrix storing all association marginals
    $\mathcal{A} \leftarrow$ HypothesisEnumeration($Z_k$)        ▷ See Algorithm 2
    for $a \in \mathcal{A}$ do
        $p \leftarrow 1$
        for $(t, j) \in a$ do
            if $j = 0$ then
                $p \leftarrow p \cdot (1 - P_d)$
            else
                $p \leftarrow p \cdot P_d l^{tj} \;/\; (P_d \nu + \lambda)$
        for $(t, j) \in a$ do
            $P_{tj} \leftarrow P_{tj} + p$
    $\mathbf{P} \leftarrow$ NormalizeMarginals($\mathbf{P}$)
    return $\mathbf{P}$
```

Algorithm 2 Explicit hypothesis enumeration.

```
procedure HypothesisEnumeration($Z_k$)
    $\mathcal{B} = \{\ \}$
    $b = \{\ \}$
    $m_k = |Z_k|$
    TraverseHypothesisTree($\mathcal{B}, b, 1, m_k$)        ▷ See Algorithm 3
    $\mathcal{A} = \{\ \}$        ▷ Convert $\mathcal{B}$ to $\mathcal{A}$
    for $b \in \mathcal{B}$ do
        $a = \{\ \}$
        for $t \in \{1, \ldots, n_k\}$ do
            if Associated($b, t$) then
                $j =$ AssociatedMeasurement($b, t$)
                $a \leftarrow a \cup \{(t, j)\}$
            else
                $a \leftarrow a \cup \{(t, 0)\}$
        $\mathcal{A} \leftarrow \mathcal{A} \cup a$
    return $\mathcal{A}$
```

Algorithm 3 Hypothesis tree traversal.

```
procedure TraverseHypothesisTree($\mathcal{B}, b, j, m_k$)
    if $j > m_k$ then                                   ▷ All measurements associated
        $\mathcal{B} \leftarrow \mathcal{B} \cup b$
        return
    $b \leftarrow b \cup \{(0, j\}$                      ▷ Add misdetection and continue down
    TraverseHypothesisTree($\mathcal{B}, b, j + 1, m_k$)
    $b \leftarrow b \setminus \{(0, j)\}$             ▷ Remove misdetection and try other branches
    for $t \in \{1, \dots, n_k\}$ do                     ▷ Explore each possible branch
        if Unassociated($b, t$) and Gated($t, j$) then
            $b \leftarrow b \cup \{(t, j\}$
            TraverseHypothesisTree($\mathcal{B}, b, j + 1, m_k$)
            $b \leftarrow b \setminus \{(t, j)\}$
    return
```

3.3.2 Finding the M best hypotheses with Murty's method

Since enumerating all possible, valid association hypotheses is in practice infeasible, a common heuristic for approximating the marginals is to *enumerate only the M hypotheses with the highest probability*, as usually the remaining hypotheses will have negligible probability [45]. The algorithm that makes this possible is called *Murty's method*, named after its inventor [46] which published the method back in 1968. The method was later adopted into the MTT community by Cox, Miller et. al in [47] which optimized the algorithm for use in MHT. Later, by Danchick and Newnam in [48], Reid's MHT method was reformulated to incorporate Murty's method.

Embedded in Murty's method is a *linear assignment solver* that solves the underlying *mutual exclusion assignment problem* between tracks and measurements which follows from the at-most-one assumptions discussed in Chapter 3.1.2. Common choices [49] are the *Hungarian method* [50], the *auction method* [51] and the *Joker-Volgenant (JV) algorithm* [52]. In [47] they used the JV algorithm to accelerate Murty's by using the dual variables from the JV algorithm as bounds for choosing an order to solve the most promising problems first and what parent tracks to process first [49].

II

Multihypothesis data association in multitarget tracking

4 | Approximating the association marginals

The following chapter is the most important in the entire report and is where the main contributions and novelties are presented. Having derived the joint association hypothesis posterior in Chapter 3, we are properly equipped to compute the necessary marginals. First, the association problem and the novel factor graph structure of the multihypothesis association hypothesis density are described, for then to introduce two novel algorithms for approximating the exact association marginals.

4.1 Factor graph representation of joint multihypothesis association hypothesis posterior

We will here present the novel factor graph representation of the joint multihypothesis association hypothesis posterior, which is based upon the work in [13], but introduces two novelties. Firstly, the *hypothesis variable* θ, which extends the inference capabilities of the factor graph to be multihypothesis. Secondly, we introduce the *nonexistence state* $a^t = N$ for all tracks t. Intuitively, this state encodes the notion that tracks are only initialized in a single, previous hypothesis, and so we can only declare tracks as misdetected or detected if they exist. More importantly, by adding the extra nonexistence state to the track association variable, we are able to compute the desired multihypothesis track existence probability that we need for track recycling in PMBM.

Based on this, the new factorization can be derived by inspecting the distribution given in (3.71) as follows. We use the same overparameterization of track-measurement associations as in [13] by introducing the association variable b^j, $j = 1, \ldots, m_k$ with m_k being the number of measurements, defined as $b^j = t$ if measurement j is associated with track t and $b^j = 0$ is measurement j is a misdetection, similarly to how we did in

Chapter 3.3.1. We then require the compatibility factors γ^{tj} between the tracks a^t and measurements b^j which are given as

$$\gamma^{tj}(a^t, b^j) = \begin{cases} 0, & a^t = j \wedge b^j \neq t \ \vee \ a^t \neq j \wedge b^j = t \\ 1, & \text{otherwise} \end{cases} \tag{4.1}$$

in order to assign 0 probability to invalid association hypotheses that disobeys the at-most-one assumption discussed in Chapter 3.1.2.

We also add the prior factor for the hypothesis variable θ, which was denoted by the factor $\mathsf{Pr}\{\theta^l_{1:k-1} | Z_{1:k-1}\}$ in (3.71), to the factorization. Here too are compatibility factors required, denoted by ζ^t, between tracks a^t and θ, defined by

$$\zeta^t(\theta, a^t) = \begin{cases} 1, & t \in \theta \wedge a^t = j, j \in \{0, 1, \ldots, m_k\} \vee t \notin \theta \wedge a^t = N \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

which we require to encode the nonexistence state $a^t = N$. The logical statement for $\zeta^t(a^t, \theta) = 1$ can be interpreted as one of two mutually exclusive requirements that must be fulfilled. One of the requirements, the *existence consistency requirement*, is that $a^t = j$, $j = 0, 1, \ldots, m_k$, i.e. a track t can only be associated with misdetection or detection in the cases where θ takes the value of a hypothesis containing track t. The alternative requirement, the *nonexistence consistency requirement*, is that $a^t = N$, i.e. track t does not exist, only in the cases when θ takes the value of a hypothesis that does not contain track t. Similarly to the association compatibility factor γ^{tj} in (4.1), the purpose of this factor is to assign 0 probability to invalid association hypotheses.

We will now rewrite the expressioncontributions in (3.71) in a factorized form that we can use to build a factor graph. Note that we have in (3.71) evaluated the hypothesis posterior in a specific parent hypothesis $\theta^r_{1:k}$ which branches of the child hypothesis $\theta^l_{1:k-1}$. To generalize the expression for all hypotheses θ we include the compatibility factor ζ^t from (4.2) and add the factors 1 for all tracks that does not exist in θ. Without adding the extra association variable b^j, this lets us arrive at the expression

$$\mathsf{Pr}\{\theta_{1:k}|Z_{1:k}\} \propto \underbrace{\mathsf{Pr}\{\theta_{1:k-1}|Z_{1:k-1}\}}_{\varphi(\theta)} \underbrace{(1-P_d)^{\bar{\delta}_k} \prod_{t:a_k^t>0} \frac{P_d l^{ta^t}}{P_d\nu+\lambda}}_{\prod_{t\in\theta}\zeta^t(\theta,a^t)\psi^t(a^t)} \cdot \underbrace{1}_{\prod_{t\notin\theta}\zeta^t(\theta,a^t)\psi^t(a^t)} \tag{4.3}$$

$$= \varphi(\theta)\prod_{t\in\theta}\zeta^t(\theta,a^t)\psi^t(a^t)\prod_{t\notin\theta}\zeta^t(\theta,a^t)\psi^t(a^t) \tag{4.4}$$

$$= \varphi(\theta)\prod_{t_1}^{n_k}\zeta^t(\theta,a^t)\psi^t(a^t) \tag{4.5}$$

where n_k denotes the number of tracks and we have introduced the short-hand notation for the prior hypothesis distribution

$$\varphi(\theta) = \mathsf{Pr}\{\theta_{1:k-1} \mid Z_{1:k-1}\}\,. \tag{4.6}$$

and the prior factor $\psi^t(a^t)$ for each track which is given by

$$\psi^t(a^t=0) = 1-P_d, \tag{4.7}$$

$$\psi^t(a^t=j) = \frac{P_d l^{tj}}{P_d\nu+\lambda}, \quad j\in\{1,\dots,m_k\} \tag{4.8}$$

$$\psi^t(a^t=N) = 1. \tag{4.9}$$

In the above derivation, the definition $\psi^t(a^t = N) = 1$ was a convenient trick to incorporate nonexisting tracks into the product without affecting the result. There is, however, a theoretical result that shows that this is indeed correct to do. The result uses *Finite set statistics* (FISST) and so it is not included here. The reader is instead referred to [14], [33].

The final step of adding b^j is done by simply adding in an extra product over all m_k measurements inside the product for each track t, arriving at

$$\mathsf{Pr}\{\theta_{1:k}|Z_{1:k}\} \propto \varphi(\theta)\prod_t\left(\zeta^t(\theta,a^t)\psi^t(a^t)\prod_{j=1}^{m_k}\gamma^{tj}(a^t,b^j)\right). \tag{4.10}$$

An illustrative example of how such a factor graph can look like can be found in Figure 4.1 for a tracking scenario where we have three tracks a^1, a^2 and a^3 and two measurements b^1 and b^2.

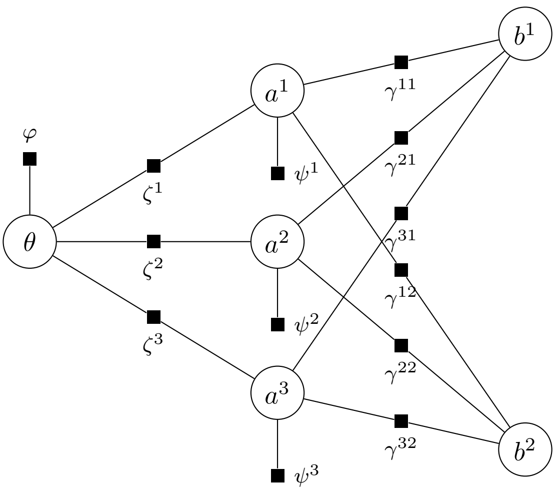


Figure 4.1: *A toy example with three tracks a^1, a^2 and a^3 and two measurements b^1 and b^2.*

4.2 Hypothesis-conditioned loopy belief propagation with probability hypothesis density approximation likelihood

Before presenting the main result of this report, we will first consider an alternative approach to marginalization of the joint multihypothesis association hypothesis posterior. We can rewrite the desired marginals as a total probability over all prior hypotheses, such that one first computes the hypothesis-conditioned marginals using LBP as in [13], for then to sum these together with appropriate scaling. By total probability and Bayes' rule, the marginal can be written as

$$\begin{aligned}\mathsf{Pr}\{a^t \mid Z_{1:k}\} &= \sum_{\theta} \mathsf{Pr}\{a^t \mid \theta, Z_{1:k}\} \mathsf{Pr}\{\theta \mid Z_{1:k}\} && (4.11)\\ &\propto \sum_{\theta} \mathsf{Pr}\{a^t \mid \theta, Z_{1:k}\} p(Z_k \mid \theta, Z_{1:k-1}) \mathsf{Pr}\{\theta \mid Z_{1:k-1}\}. && (4.12)\end{aligned}$$

For tracks that exist in the prior hypothesis θ, the marginal $\mathsf{Pr}\{a^t \mid \theta, Z_{1:k}\}$ can be computed with LBP, setting $\mathsf{Pr}\{a^t = N \mid \theta, Z_{1:k}\} = 0$. For tracks that does not exist in the prior hypothesis we set $\mathsf{Pr}\{a^t = N \mid \theta, Z_{1:k}\} = 1$ and all other association events to 0. The prior probability $\mathsf{Pr}\{\theta \mid Z_{1:k-1}\} = \varphi(\theta)$ is assumed given, and so using this approaches reduces to computing the hypothesis-conditioned set likelihood $p(Z_k \mid \theta, Z_{1:k-1})$. Computing it exactly involves full hypothesis enumeration, which is in general infeasible. It can, however, be approximated using *random finite set theory*, namely, by using approximations from the *Probability hypothesis density* (PHD) filter [53]. Therefore, going forward we will refer to this method by the name *Hypothesis-conditioned loopy belief propagation with PHD approximation likelihood* (LBP-PHD). Intuitively, we can think of the true $p(Z_k|\theta, Z_{1:k-1})$ as a binomial distribution over the possible measurements we can get, considering both detections and clutter. Similarly to what we did for clutter in Chapter 3.1.2 we can therefore approximate it with a Poisson distribution. The exact derivation details are outside the scope of this text. The result is that we can approximate the hypothesis-conditioned likelihood with

$$p(Z_k|\theta, Z_{1:k-1}) \approx e^{-\sum_{t=1}^{n_k} r_t P_d} \prod_{j=1}^{m_k} \left[\left(\sum_{t=1}^{n_k} \frac{r_t P_d l^{tj}}{P_d \nu + \lambda} \right) + 1 \right] \tag{4.13}$$

where we can recognize the misdetection probabilities in the sum in the exponent by rewriting as

$$\sum_{t=1}^{n_k} r_t P_d = \sum_{t=1}^{n_k} 1 - (1 - r_t P_d), \tag{4.14}$$

the detection likelihoods from (3.71) in the product and the 1 accounts for the fact that a measurement can be clutter and is equal to 1 as we have normalized the detection likelihood by $P_d \nu + \lambda$.

4.3 Multihypothesis loopy belief propagation

The following section is the main, novel theoretical result of this report and will introduce the notion of approximating the desired marginals by use of the LBP that was first described in Chapter 2.4. The method will be built upon the work presented in [13], which presents an association graph with similar structure to the one presented in Figure 4.1, but without the presence of a prior hypothesis variable θ and the nonexistence state $a^t = N$. Thus, in the following, this extension of LBP will be referred to as *Multihypothesis loopy belief propagation* (MH-LBP). There are four types of messages that are sent in the graph. The message sent from a track t to a measurement j is denoted by $\mu_{t \to j}$, the message sent from a measurement j to a track t is denoted by $\nu_{j \to t}$, the message from the prior hypothesis θ to a track t is denoted by σ_t and finally, the message from a track t to the prior hypothesis θ is denoted by ρ_t. The message definitions are summarized in Table 4.1 and their directions illustrated in Figure 4.2.

In principle, doing LBP is matter of computing the messages

$$\mu_{a\to i}(x_i) \leftarrow \sum_{x_{\mathsf{N}(a)\setminus\{i\}}} f_a(x_{\mathsf{N}(a)}) \prod_{j\in\mathsf{N}(a)\setminus\{i\}} \mu_{j\to a}(x_j), \tag{4.15}$$

$$\mu_{i\to a}(x_i) \leftarrow \prod_{b\in\mathsf{N}(i)\setminus\{a\}} \mu_{b\to i}(x_i) \tag{4.16}$$

repeatedly until convergence, where the equations (4.15) and (4.16) are the same as in (2.23) and (2.24) and repeated here for convenience. By inserting the factors (4.1), (4.2) and (4.6) to (4.9) that we defined in Chapter 4.1 into (4.15) and (4.16) and using the message notation from Table 4.1, the general LBP equations take the form

$$\mu_{t\to j}(b^j) = \sum_{a^t} \psi^t(a^t)\gamma_{tj}(a^t, b^j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t)\right) \sigma_t(a^i), \tag{4.17}$$

$$\nu_{j\to t}(a^t) = \sum_{b^j} \gamma^{tj}(a^t, b^j) \prod_{t'\neq t} \mu_{t'\to j}(b^j), \tag{4.18}$$

$$\sigma_t(a^t) = \sum_{\theta} \zeta^t(\theta, a^t)\varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta), \tag{4.19}$$

$$\rho_t(\theta) = \sum_{a^t} \zeta^t(\theta, a^t)\psi_t(a^t) \prod_{j} \nu_j(a^t), \tag{4.20}$$

where $\sum_{b^j}$ denotes the sum over all values $b^j \in \{0, 1, \ldots, n_k\}$, $\sum_{a^t}$ denotes the sum over all values $a^t \in \{0, 1, \ldots, m_k, N\}$, $\sum_\theta$ denotes the sum over all values $\theta \in \{\theta^1, \ldots, \theta^L\}$ for L prior hypotheses, $\prod_{j'\neq j}$ denotes the product over all measurements except for the j^{th}, $\prod_{t'\neq t}$ denotes the product over all tracks except for the t^{th} and $\prod_j$ is the product over all measurements. The key insight is that all messages have similar behavior to what is recognized in [13], which allows for clever normalizations for reducing computation complexity and simpler expressions. This is because we can show that, although the messages above are strictly speaking functions of a^t, b^t and θ, *we can use the structure of the graph to reduce the messages to scalar values instead of tables of values*. This takes less resources to compute and store in memory, which has great benefits when implementing and executing the algorithm. The result is in Theorem 1, which follows.

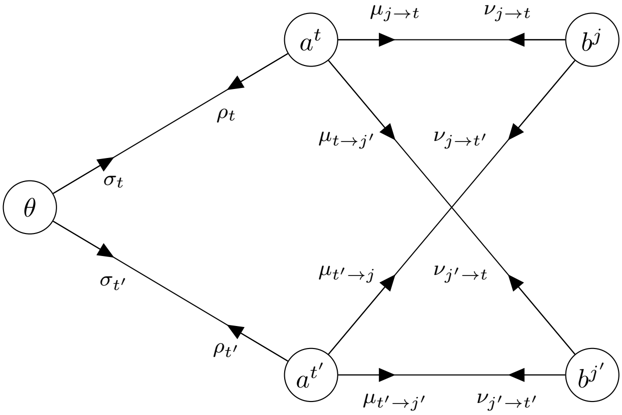


Figure 4.2: *Simplified illustraion of message directions in association graph.*

Name	Notation	Direction
Track-to-measurement	$\mu_{t\to j}$	$a^t \to b^j$
Measurement-to-track	$\nu_{j\to t}$	$b^j \to a^t$
Hypothesis-to-track	σ_t	$\theta \to a^t$
Track-to-hypothesis	ρ_t	$a^t \to \theta$

Table 4.1: *Message types in association graph.*

Theorem 1 (The message definitions for multihypothesis LBP)**.** *Given an association graph of the same structure as in Figure 4.1 where the factors are defined as in* (4.1)*,* (4.2) *and* (4.6) *to* (4.9)*, the normalized messages used in multihypothesis LBP are given as*

$$\mu_{t\to j} = \frac{\psi^t(j)}{\psi^t(0) + \sum_{j'\neq j, j'>0} \psi^t(j')\nu_{j'\to t} + \sigma_t}, \tag{4.21a}$$

$$\nu_{j\to t} = \frac{1}{1 + \sum_{t'\neq t, t'>0} \mu_{t'\to j}}, \tag{4.21b}$$

$$\sigma_t = \rho_t \cdot \frac{\sum\limits_{\theta\,:\,t\notin\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}{\sum\limits_{\theta\,:\,t\in\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}, \tag{4.21c}$$

$$\rho_t = \psi^t(0) + \sum_{j=1}^{m_k} \psi^t(j)\nu_{j\to t}. \tag{4.21d}$$

Proof. We will first simplify the track-to-measurement message $\mu_{t\to j}$ as much as possible at this point. The sum is over all values of a^t, j included, so we first explicitly separate the sum into the term where $a^t = j$ and a partial sum over the remaining a^t as

$$\begin{aligned}\mu_{t\to j}(b^j) = \psi_t(a^t = j)\gamma_{tj}(a^t = j, b^j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t)\right) \sigma_t(a^t) \\ + \sum_{a^t\neq j} \psi_t(a^t)\gamma_{tj}(a^t \neq j, b^j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t)\right) \sigma_t(a^t). \qquad (4.22)\end{aligned}$$

By now inserting $b^j = t$, we see that

$$\sum_{a^t\neq j} \psi_t(a^t)\gamma_{tj}(a^t \neq j, b^j = t) \prod_{j'\neq j} \nu_{j'\to t} = 0 \qquad (4.23)$$

and

$$\psi_t(a^t = j)\gamma_{tj}(a^t = j, b^j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t)\right) \sigma_t(a^t) = \psi_t(a^t = j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t)\right) \sigma_t(a^t) \qquad (4.24)$$

as $\gamma_{tj}(a^t \neq j, b^j = t) = 0$ and $\gamma_{tj}(a^t = j, b^j = t) = 1$, respectively, by the way it was defined in (4.1). Doing the same for $b^j \neq t$ gives

$$\psi_t(a^t = j)\gamma_{tj}(a^t = j, b^j \neq t) \prod_{j'\neq j} \nu_{j'\to t}(a^t)\sigma_t(a^t) = 0 \qquad (4.25)$$

and

$$\sum_{a^t\neq j} \psi_t(a^t)\gamma_{tj}(a^t \neq j, b^j = t) \prod_{j'\neq j} \nu_{j'\to t} = \sum_{a^t\neq j} \psi_t(a^t) \prod_{j'\neq j} \nu_{j'\to t} \qquad (4.26)$$

for similar reasons. Thus, we get that the message value reduces to two distinct values,

$$\mu_{t\to j}(b^j) = \begin{cases} \psi_t(a^t = j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t = j)\right) \sigma_t(a^t = j), & b^j = t \\ \sum_{a^t\neq j} \psi_t(a^t) \prod_{j'\neq j} \nu_{j'\to t}(a^t)\sigma_t(a^t), & b^j \neq t. \end{cases} \qquad (4.27)$$

Since messages in LBP are only given up to scale, we can normalize them. Namely, by normalizing $\mu_{t\to j}$ by its value when $b^j \neq t$, we get that

$$\mu_{t\to j}(b^j = t) = \frac{\psi_t(a^t = j)\prod_{j'\neq j}\nu_{j'\to t}(a^t)\sigma_t(a^t)}{\sum_{a^t\neq j}\psi_t(a^t)\prod_{j'\neq j}\nu_{j'\to t}(a^t)\sigma_t(a^t)}, \tag{4.28}$$

$$\mu_{t\to j}(b^j \neq t) = 1. \tag{4.29}$$

For now, these are the simplifications we can do. The expression in (4.28) will be further simplified later.

We now consider the measurement-to-track message $\nu_{j\to t}$. We start by doing the same as for $\mu_{t\to j}$ above by explicitly separating the sum into the term where $b^j = t$ and the partial sum where $b^j \neq t$ to get

$$\begin{aligned}\nu_{j\to t}(a^t) = \gamma_{tj}(a^t, b^j = t)\prod_{t'\neq t}\mu_{t'\to j}(b^j = t)\\ + \sum_{b^t\neq t}\gamma_{tj}(a^t, b^j)\prod_{t'\neq t}\mu_{t'\to j}(b^j).\end{aligned} \tag{4.30}$$

We can then reduce the message value to the two distinct values

$$\nu_{j\to t}(a^t) = \begin{cases}\prod_{t'\neq t}\mu_{t'\to j}(b^j = t), & a^t = j\\ \sum_{b^t\neq t}\prod_{t'\neq t}\mu_{t'\to j}(b^j), & a^t \neq j.\end{cases} \tag{4.31}$$

by following a similar line of reasoning as for $\mu_{t\to j}$. We choose to normalize by $\nu_{j\to t}(a^t \neq j)$ to get

$$\nu_{j\to t}(a^t = j) = \frac{\prod_{t'\neq t}\mu_{t'\to j}(b^j = t)}{\sum_{b^t\neq t}\prod_{t'\neq t}\mu_{t'\to j}(b^j \neq t)}, \tag{4.32}$$

$$\nu_{j\to t}(a^t \neq j) = 1. \tag{4.33}$$

If we now insert (4.29) into (4.32) we get that the numerator reduces to

$$\prod_{t'\neq t}\mu_{t'\to j}(b^j = t) = \prod_{t'\neq t} 1 \tag{4.34}$$

$$= 1 \tag{4.35}$$

and the denominator becomes

$$\sum_{b^t \neq t} \prod_{t' \neq t} \mu_{t' \to j}(b^j \neq t) = \prod_{t' \neq t} \mu_{t' \to j}(b^j = 0) + \sum_{\substack{t''>0 \\ t'' \neq t}} \prod_{t' \neq t} \mu_{t' \to j}(b^j = t''), \tag{4.36}$$

$$= \prod_{t' \neq t} 1 + \sum_{\substack{t''>0 \\ t'' \neq t}} \left(\mu_{t'' \to j}(b^j = t'') \prod_{\substack{t' \neq t \\ t' \neq t''}} 1 \right), \tag{4.37}$$

$$= 1 + \sum_{\substack{t''>0 \\ t'' \neq t}} \mu_{t'' \to j}(b^j = t''), \tag{4.38}$$

which, after changing back the dummy variable t'' to t' in (4.38), gives the final expression

$$\nu_{j \to t} = \frac{1}{1 + \sum_{t' \neq t, t' > 0} \mu_{t' \to j}}, \tag{4.39}$$

which is the same as in (4.21b).

Next we turn to the hypothesis-to-track message σ_t. If we first rewrite the sum in (4.19) as the sum of two partial sums,

$$\sigma_t(a^t) = \sum_{\theta :\, t \in \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) + \sum_{\theta :\, t \notin \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta), \tag{4.40}$$

where the notation $\theta :\ t \in \theta$ and $\theta :\ t \notin \theta$ means all prior hypotheses θ containing and not containing the track t, respectively. We again apply a similar procedure as for $\mu_{t \to j}$ and $\nu_{j \to t}$, only this time $\zeta_t(\theta, a^t)$ takes the role of $\gamma_{tj}(a^t, b^j)$. For $a^t \in \{0, 1, \ldots, m_k\}$ and $t \in \theta$

$$\sum_{\theta :\, t \notin \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) = 0 \tag{4.41}$$

and

$$\sum_{\theta :\, t \in \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) = \sum_{\theta :\, t \in \theta} \varphi(\theta) \prod_{t' \neq t} \rho_{t'}(\theta) \tag{4.42}$$

as $\zeta_t(\theta, a^t) = 0$ and $\zeta_t(\theta, a^t) = 1$, respectively, by the way it was defined in (4.2). Similarly, when $a^t = N$ and $t \in \theta$,

$$\sum_{\theta :\, t \in \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) = 0 \tag{4.43}$$

and

$$\sum_{\theta:\, t\notin\theta} \varphi(\theta)\zeta_t(\theta, a^t) \prod_{t'\neq t} \rho_{t'}(\theta) = \sum_{\theta:\, t\notin\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta), \tag{4.44}$$

Thus, σ_t reduces to the two cases

$$\sigma_t(a^t) = \begin{cases} \sum_{\theta:\, t\in\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta), & a^t = 0, 1, \dots, m_k \\ \sum_{\theta:\, t\notin\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta), & a^t = N. \end{cases} \tag{4.45}$$

We choose to normalize by $\sigma_t(a^t \neq N)$ to get the values

$$\sigma_t(a^t = N) = \frac{\sum\limits_{\theta:\, t\notin\theta} \varphi(\theta) \prod\limits_{t'\neq t} \rho_{t'}(\theta)}{\sum\limits_{\theta:\, t\in\theta} \varphi(\theta) \prod\limits_{t'\neq t} \rho_{t'}(\theta)}, \tag{4.46}$$

$$\sigma_t(a^t \neq N) = 1. \tag{4.47}$$

We will return to (4.46) soon. First, we will return to the expression for the track-to-measurement message $\mu_{t\to j}$, as we have all the pieces we need to simplify the message in (4.28). Inserting (4.33) and (4.47) into (4.28) makes the numerator

$$\psi_t(a^t = j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t = j) \right) \sigma_t(a^t = j) = \psi_t(a^t = j) \left(\prod_{j'\neq j} 1 \right) \cdot 1 \tag{4.48}$$

$$= \psi_t(a^t = j) \tag{4.49}$$

and the denominator

$$\begin{aligned}\sum_{a^t \neq j} \psi^t(a^t) \left(\prod_{j' \neq j} \nu_{j' \to t}(a^t) \right) \sigma_t(a^t) &= \psi^t(a^t = 0) \left(\prod_{j' \neq j} \nu_{j' \to t}(a^t = 0) \right) \sigma_t(a^t = 0) \\ &+ \sum_{\substack{a^t = 1 \\ a^t \neq j}}^{m_k} \psi^t(a^t) \left(\prod_{j' \neq j} \nu_{j' \to t}(a^t) \right) \sigma_t(a^t) \\ &+ \psi^t(a^t = N) \left(\prod_{j' \neq j} \nu_{j' \to t}(a^t = N) \right) \sigma_t(a^t = N) \end{aligned} \tag{4.50}$$

$$\begin{aligned} &= \psi^t(a^t = 0) \left(\prod_{j' \neq j} 1 \right) \cdot 1 \\ &+ \sum_{\substack{a^t = 1 \\ a^t \neq j}}^{m_k} \psi^t(a^t) \nu_{a^t \to t}(a^t) \left(\prod_{\substack{j' \neq j \\ j' \neq a^t}} 1 \right) \cdot 1 \\ &+ 1 \cdot \left(\prod_{j' \neq j} 1 \right) \sigma_t(a^t = N) \end{aligned} \tag{4.51}$$

$$= \psi^t(0) + \sum_{j' \neq j, j' > 0} \psi^t(j') \nu_{j' \to t} + \sigma_t \tag{4.52}$$

where we used that $\psi^t(a^t = N) = 1$ from (4.9). Putting it back together we get

$$\mu_{t \to j} = \frac{\psi_t(j)}{\psi_t(0) + \sum_{j' \neq j, j' > 0} \psi_t(j') \nu_{j' \to t} + \sigma_t}. \tag{4.53}$$

which again is the desired result in (4.21a).

The track-to-hypothesis message $\rho_t(\theta)$ can be simplified as follows. We do a decomposition of the sum in (4.20) into a partial sum over $a^t = 0, 1, \ldots, m_k$ and the term for $a^t = N$ to get

$$\rho_t(\theta) = \sum_{a^t \neq N} \psi^t(a^t) \zeta^t(\theta, a^t) \prod_j \nu_{j \to t}(a^t) + \psi^t(N) \zeta^t(\theta, a^t = N) \prod_j \nu_{j \to t}(N). \tag{4.54}$$

Performing the same procedure as for σ_t above, we get that inserting θ when $t \in \theta$ makes

$$\psi^t(N)\zeta^t(\theta, a^t = N)\prod_j \nu_{j\to t}(N) = 0 \tag{4.55}$$

and

$$\sum_{a^t \neq N} \psi^t(a^t)\zeta^t(\theta, a^t)\prod_j \nu_{j\to t}(a^t) = \sum_{a^t \neq N} \psi^t(a^t)\prod_j \nu_{j\to t}(a^t) \tag{4.56}$$

due to $\zeta^t(\theta, a^t) = 0$ and $\zeta^t(\theta, a^t) = 1$, respectively, while for θ when $t \notin \theta$ makes

$$\sum_{a^t \neq N} \psi^t(a^t)\zeta^t(\theta, a^t)\prod_j \nu_{j\to t}(a^t) = 0 \tag{4.57}$$

and

$$\psi^t(N)\zeta^t(\theta, a^t = N)\prod_j \nu_{j\to t}(N) = \psi^t(N)\prod_j \nu_{j\to t}(N) \tag{4.58}$$

for similar reasons. Consequently, as before, the message reduces to two cases,

$$\rho_t(\theta) = \begin{cases} \sum_{a^t \neq N} \psi_t(a^t)\prod_j \nu_j(a^t), & t \in \theta \\ \psi_t(N)\prod_j \nu_j(N). & t \notin \theta \end{cases} \tag{4.59}$$

By inserting $\psi^t(N) = 1$ from (4.9) and $\prod_j \nu_j(N) = 1$ from (4.33) we get that the $t \notin \theta$ case is equal to 1, hence no normalization is necessary in this case. If we separate the term for $a^t = 0$ the $t \in \theta$ case becomes

$$\rho_t = \psi^t(0) + \sum_{j=1}^{m_k} \psi^t(j)\nu_{j\to t} \tag{4.60}$$

which we recognize as (4.21d).

The only thing that remains is to simplify (4.46). Note that the product $\prod_{t'\neq t} \rho_{t'}$ in the numerator can be written as $\prod_{t'} \rho_{t'}$, i.e. over all tracks, as $\rho_t = 1$ for all terms in that sum. We can further reduce the number of factors to $\prod_{t'\in\theta} \rho_{t'}$ by normalization. For the product in the denominator we do the same trick, only we now need to divide by ρ_t

as well as it is no longer unity. Thus, the final message definition used is

$$\sigma_t = \frac{\sum\limits_{\theta\colon t\notin\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}{\frac{1}{\rho_t} \sum\limits_{\theta\colon t\in\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}$$

$$= \rho_t \cdot \frac{\sum\limits_{\theta\colon t\notin\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}{\sum\limits_{\theta\colon t\in\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}. \tag{4.61}$$

The benefit of this is that this allows for reusing of computation and lower overall complexity by computing $\prod_{t'\in\theta} \rho_{t'}$ for each θ before computation of σ_t. ■

We can now run LBP using these messages. After convergence, the approximate association marginals can be computed from

$$\hat{p}(a^t|Z_{1:k}) \propto \begin{cases} \psi^t(0), & a^t = 0 \\ \psi^t(j)\nu_{j\to t}, & a^t = 1,\dots,m_k \\ \sigma_t, & a^t = N \end{cases} \tag{4.62}$$

while the measurement marginals are computed with

$$\hat{p}(b^j|Z_{1:k}) \propto \begin{cases} 1, & b^j = 0, \\ \mu_{t\to j}, & b^j = 1,\dots,n_k \end{cases} \tag{4.63}$$

and the prior hypothesis posterior

$$\hat{p}(\theta|Z_{1:k}) \propto \varphi(\theta) \prod_{t\in\theta} \rho_t \tag{4.64}$$

4.3.1 Algorithmic complexity

In [13] they argue that the computation of $\mu_{t\to j}$ and $\nu_{j\to t}$ is $\mathcal{O}(n_k m_k)$, which holds here as well. The computation of ρ_t for a given t consists of a sum which is m_k large, which over n_k targets totals in $\mathcal{O}(n_k m_k)$ computations, hence not changing the complexity. For a given σ_t, we need to compute H_k terms, with H_k denoting number of prior hypotheses at timestep k, that makes up the total number of terms in the partial sums that appear in the numerator and denominator. Since the same products can be used for all tracks, as these only depend on the hypotheses, they can be computed beforehand in $\mathcal{O}(H_k n_k)$ time. Computation of all σ_t is then also $\mathcal{O}(H_k n_k)$, which in total means that all messages can be computed in $\mathcal{O}(m_k n_k + H_k n_k)$ time and with $\mathcal{O}(m_k n_k)$ memory, as we need $m_k n_k$ memory for $\mu_{t \to j}$ and $\nu_{j \to t}$ and n_k memory for ρ_t and σ_t.

III

Simulation results

5 | Simulation of multitarget tracking scenarios

The proposed methods for approximate marginals presented in Chapter 4 were tested on a large, simulated dataset. The simulated dataset is called “9 ravens” and the scenario consists of 1397 simulated radar scans in 2 dimensions. There are 8 true targets, while the radar is mounted on yet another vehicle. The dataset consists of 10001 timesteps of simulation. For each timestep, the methods are tested separately on each *cluster* of tracks, where each cluster is defined as a set of tracks where each track in the set has gated a measurement gated by at least one other track in the set. Extracting the cluster data from the timestep data showed that there are in total 111887 clusters in the dataset.

In order to compute estimation errors, computing the exact marginals are required, which involves explicit hypothesis enumeration. The hypothesis enumeration failed in 34 cases, equal to 0.03% of the available data, as the estimated number of hypotheses to enumerate was too great, making it impossible to compute the estimation error. This was deemed a negligible amount of the available data, and so it was simply discarded.

5.1 Overview of track clusters statistics used for testing

In Figure 5.1 there are four histograms intended to summarize the most important statistics about the clusters that the algorithms were tested on from a data association standpoint. “Number of tracks” and “number of hypotheses” refer to the number of tracks and prior hypotheses in the cluster, respectively. “Number of gated measurements” refers to the total number of measurements that were gated by tracks in a cluster. “Highest number of tracks competing for measurement” refers to, conditioned on tracks existing in a prior hypothesis, the maximum number of tracks that gated the same measurement.

There are two main takeaways from these histograms. The first, from inspecting "Number of tracks" and "Number of hypotheses", is that for the majority of clusters, there are few hypotheses and tracks, simplifying the association problem and makes approximate methods almost identical to the exact solution. This will be made more clear when we later inspect the distribution of the estimation errors. The second takeaway can be seen from inspecting "Number of gated measurements" and "Highest number of tracks competing for measurement". The more tracks that gate a measurement, the more hypotheses there are to enumerate, and so this can be thought of as a rough measure of how complicated the association problem is expected to be. As we can see, most of the association problems we encounter have few tracks competing for the same measurement.

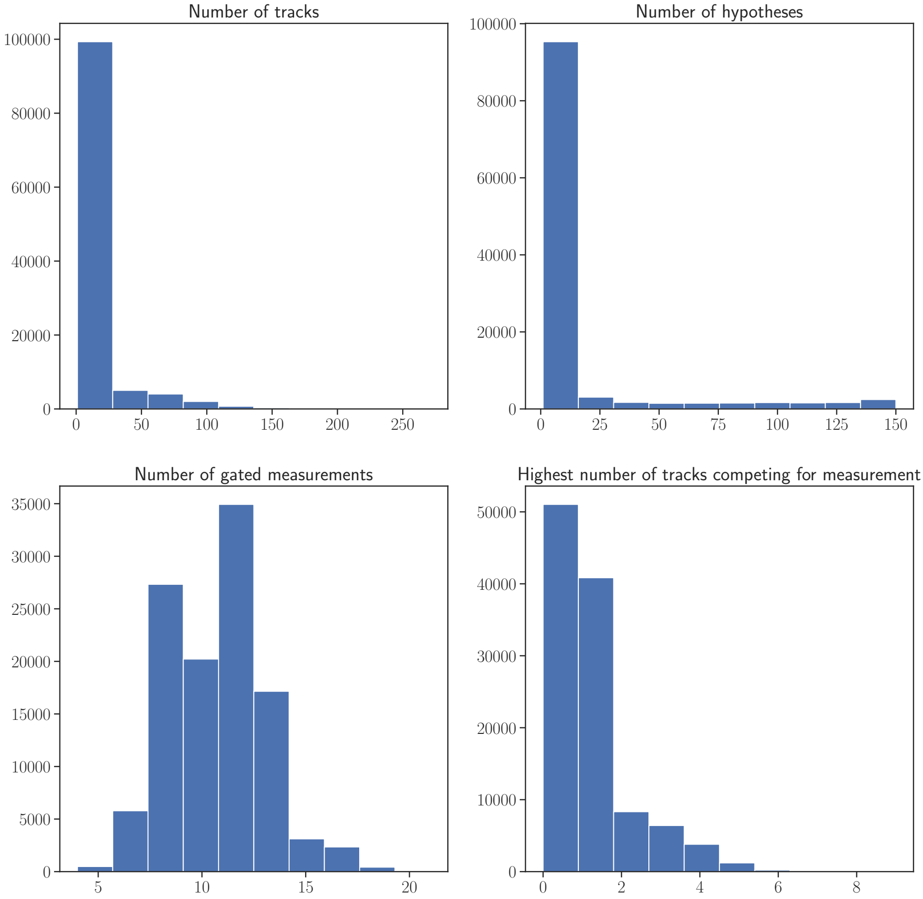


Figure 5.1: *Histograms over number of tracks, number of.*

5.1.1 Implications of number of competing tracks from a graphical point of view

From a graphical point of view, the fewer tracks that compete for the same measurement, the less correlated the tracks are to each other, and the better we expect LBP to perform. Intuitively, when we associate a track to a measurement in the association graph, we "weigh" the edge connecting them by the detection likelihood. This implies that when two tracks gate the same measurement that have commensurate detection likelihood, we get strong correlation between the tracks and and a tight cycle in the graph. On the other hand, when a track gates a measurement alone or competes about it with another track with low detection likelihood, the edge weight closing the loop gets low. Thus, we get tracks that are only weakly correlated and a loose cycle in the graph such that the graph more closely resembles a tree. Therefore, it is natural to conclude that for association problems with few tracks competing for the same measurement, LBP is better at resolving the association problem since the tree approximation it makes is more correct. In other words, for the given dataset, we have reason to believe that for most of the cases, LBP will perform well. An illustrative example comparing the two cases can be found in Figure 5.2.

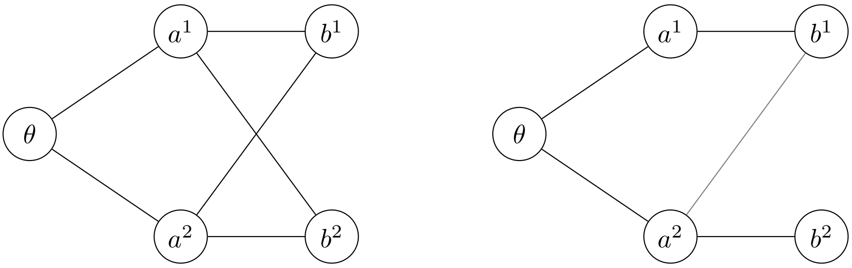


Figure 5.2: *Two different association cases. The left case is a hard case, as the two tracks a^1 and a^2 are both competing for the measurements b^1 and b^2 with close to equal detection likelihood, making two strong cycles in the graph. In the right case, however, a^2 is the only track that gates b^2 with a weak link, corresponding to low detection likelihood, to b^1, such that graph more resembles a tree, making the LBP approximation closer to exact.*

5.2 The methods compared

Three methods are compared in the following results. The two first methods are the methods MH-LBP and LBP-PHD, presented in Chapter 4.3 and Chapter 4.2, respectively. Additionally, as a benchmark or best-case to compare with, a hypothesis-conditioned LBP similar to LBP-PHD was also tested that used the exact normalization constant instead of the PHD approximation. The purpose of this was to isolate out errors from the approximate normalization in order to better capture the properties and failure modes of LBP on the multihypothesis association problem in question.

6 | Results and discussion

The following chapter will present and discuss the results of running the different marginal estimation methods on the dataset introduced in Chapter 5.

6.1 Signed marginal error and bias

First, we will discuss the histograms of the signed errors of the different methods compared, which can be found in Figure 6.1. The main reason for inspecting the signed error is to look for any bias in the estimated errors. Considering the symmetric shape of the histograms about 0 and the computed statistics in Table 6.1, we can assume that all the different methods are indeed unbiased.

Method	Signed probability error mean ± standard deviation
MH-LBP	$3.2767 \cdot 10^{-20} \pm 0.0168$
LBP-PHD	$1.1526 \cdot 10^{-20} \pm 0.0141$
LBP with exact normalization constant	$-1.0956 \cdot 10^{-21} \pm 0.0030$

Table 6.1: *Summary of signed errors statistics for the different inference methods tested on the simulated dataset.*

We can also see from Figure 6.1 that MH-LBP has a lower, upper bound on error compared to LBP-PHD, and that LBP with exact normalization constant clearly outperforms both former methods. A possible explanation for why MH-LBP has a lower error bound than LBP-PHD is that both methods inherit inaccuracies from LBP, but that LBP-PHD accumulates extra error from also having to estimate the normalization constant. If we compare LBP-PHD with LBP with exact normalization constant, we can attribute the difference in error to the estimated normalization constant, suggesting that estimating it accurately will significantly improve estimation accuracy and that hypothesis-conditioned LBP shows promise.

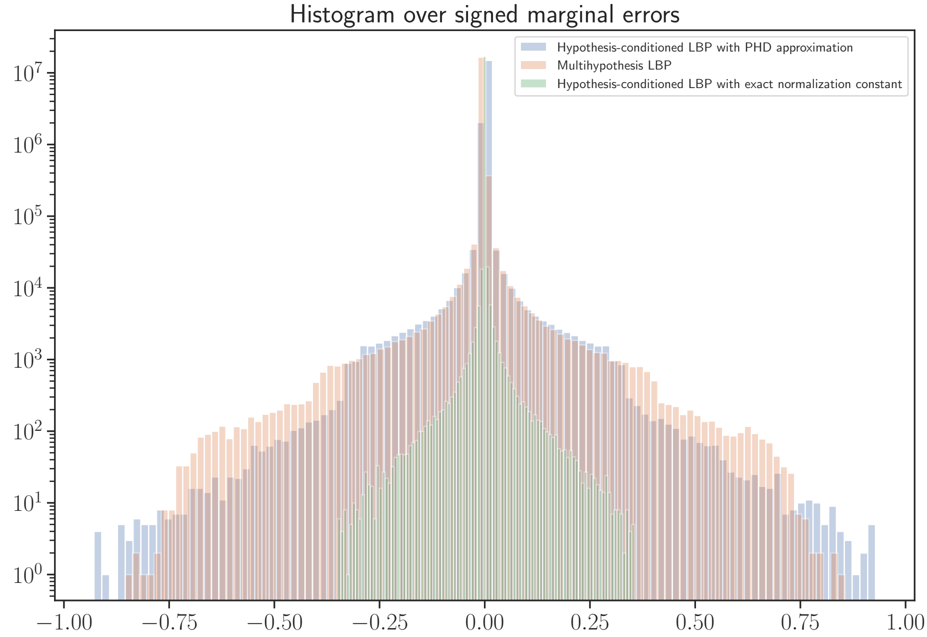


Figure 6.1: *Histogram of the signed errors between MH-LBP and hypothesis-conditioned LBP with PHD approximation. Note that the y-axis is logarithmic*

6.2 Survival function over association errors

The plots in Figure 6.2 show the *survival function*, which is simply defined as $1 - \hat{P}(e)$ with $\hat{P}(e)$ being the empirical cumulative distribution for the errors, of the empirical distribution of the errors from MH-LBP, LBP-PHD and hypothesis-conditioned LBP using the exact normalization constant as a best case to compare with. The reason we are interested in the survival function is that we wish to compare the densities of errors of the different methods, similarly to for a histogram. The main benefits of using a survival function is that we can easily compare the estimation performance of the different probabilities we are interested in and also quickly tell the best method by look at what graph hits the x-axis first. Comparing the performance of MH-LBP with that of LBP-PHD we see that overall, the two approaches are similar. Notably, MH-LBP seems to perform better in particular for misdetections than LBP-PHD, and somewhat worse for detections.

Even more interesting is how much better the LBP with exact normalization constant is at estimating the nonexistence probability. This is most likely related to how it is computed, as the hypothesis-conditioned marginals where a track does not exist is concentrated with probability 1 for nonexistence. More importantly, the crucial distinction is that we know the hypothesis-conditioned marginals for nonexisting tracks exact, while the existing tracks have only approximate marginals for misdetection and detection from LBP. Thus, the terms for nonexisting tracks in the total probability sum are also exact, and so the errors we see must be come from the LBP approximation. In other words, before renormalization of the marginals we can conclude that the unnormalized nonexistence probability is exact, but that the renormalization injects error intdataseto nonexistence probability from the remaining probabilities estimated from LBP. The most profound implication of the accurate nonexistence probability estimate is that this shows great promise in how accurate track recycling can be done in PMBM given an accurate estimate of the hypothesis-conditioned likelihood.

As a final observation, although the nonexistence probability is very exact, the misdetection and detection probabilities do not show the same behavior. As these are inferred from LBP, this is natural, as we have no guarantees about the accuracy contrary i the same way as we have for nonexistence.

6.3 Normalization constant accuracy

In Figure 6.3 the PHD approximation normalization constant is compared to the exact normalization constant in a correlation plot with logarithmic scale. In a logarithmic plot the correlation between normalization constants is better captured, mostly because the exact normalization constant in some cases is so significantly larger than the PHD estimate. Interestingly, the correlation plot demonstrates the cost of using a Poisson approximation to the measurement set over the binomial. This can be seen from the fact that for low likelihoods, the Poisson approximation overestimates the exact likelihood, while for high likelihoods it underestimates it, clearly showing the flatness of the Poisson distribution compared to the binomial. In any case, although the order of magnitude varies a lot, we can still conclude that the PHD approximation does somewhat correlate with the true normalization constant.

6.4 Correlation between approximate and exact marginals

The correlations between the exact marginals and MH-LBP, LBP-PHD and LBP with exact normalization constants, respectively, can be found in Figure 6.4.

In the correlation plot for MH-LBP most marginals are well correlated with the exact

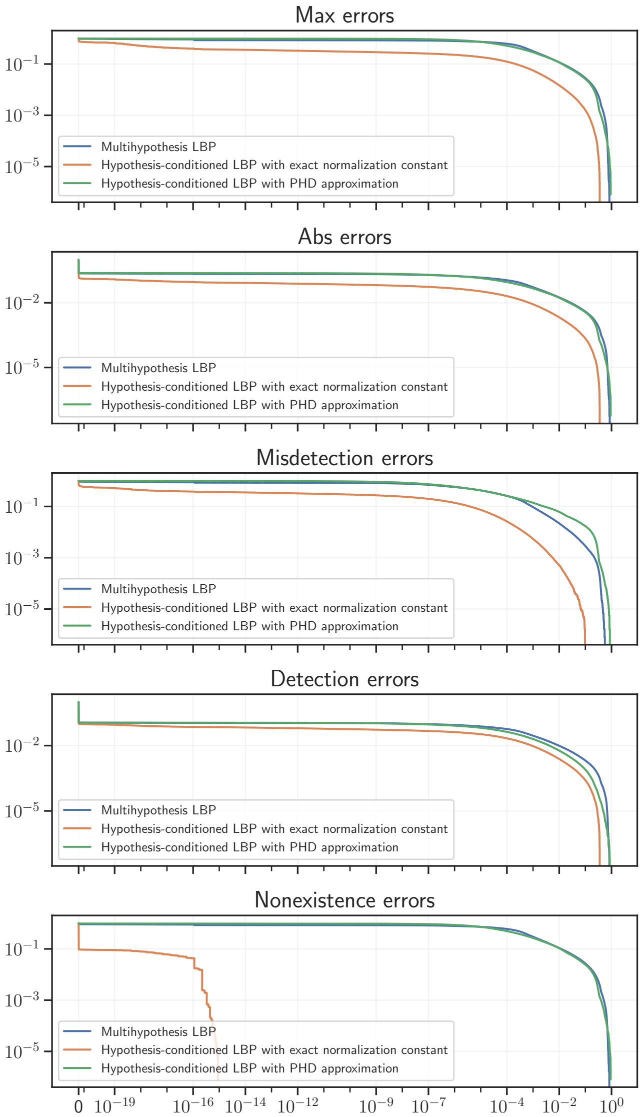


Figure 6.2: *Survival function for different errors. Note that both the y-axis and x-axis are logarithmic. The vertical line at $x = 0$ is due to all the marginals that estimated with zero error.*

marginals. However, we can clearly see an S-shaped curve that follows the point cloud of marginals. We can primarily make two observations from this. The first is that MH-LBP has a tendency of estimating individual probabilities centered at 0.5, as the density of points increases at marginals for this value, over all values of exact marginals. We can interpret this as the fact that while the exact solution would be certain, e.g. to associate

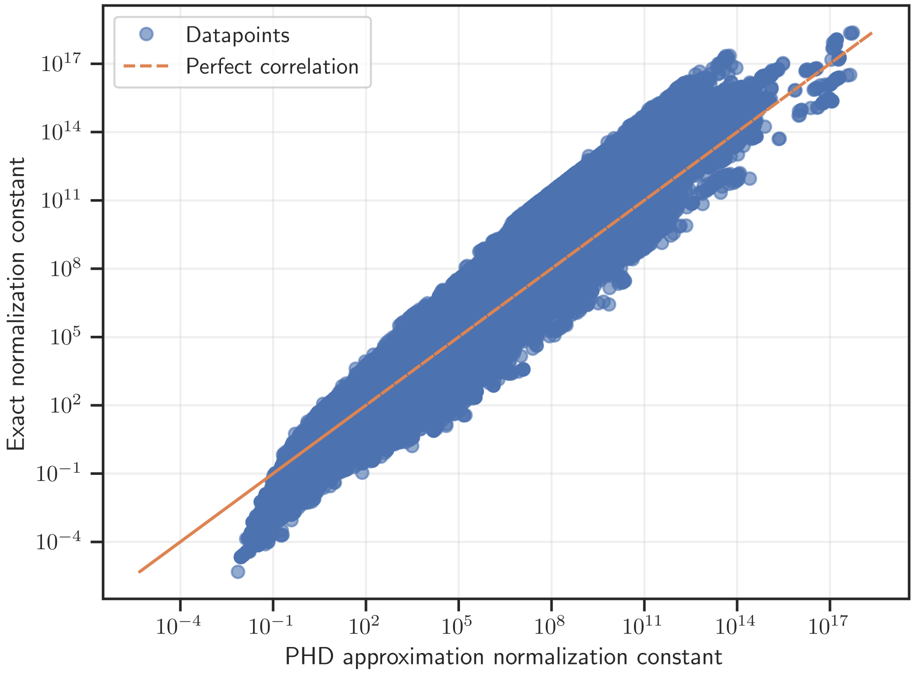


Figure 6.3: *Correlation plot between estimated normalization constant and true normalization constant with logarithmic scaling.*

with probability 0.95 or not as the association probability is only 0.05, MH-LBP would be more uncertain. This takes us to the second conclusion. For probabilities roughly below 0.5, MH-LBP tends to overestimate the probabilities as the point density is centered below the correlation line. Similarly for probabilities above 0.5 we see the opposite effect, concluding that here MH-LBP underestimates the probabilities. To conclude, it can seem from the correlation plot for MH-LBP that MH-LBP overall has a tendency to "squish" the true marginal distribution together, or at least capture the shape of it. This follows from the observation that while most estimated probabilities are well correlated with the exact probabilities, small probabilities are usually estimated to be larger than they actually are and the opposite for large probabilities.

The correlation plot for LBP-PHD shows a clear correlation line, much in the same way as for MH-LBP, but with large, convex-shaped variance about the correlation line. We also note that LBP-PHD has more data points spread across the entire plot, while MH-LBP is relatively more centered around the correlation line. A possible explanation for this can be the same as used for Figure 6.3 above, i.e., the PHD approximation. Since the Poisson approximation is flatter, tending to overestimating low likelihoods and underestimating high likelihoods, a possible explanation could be that the conservative behavior that we saw in MH-LBP also applies for hypothesis-conditioned LBP, and that this is further amplified by the PHD approximate normalization constant.

Lastly, we will inspect the correlation plot for the best-case LBP with exact normalization constant. Overall, the estimated probabilities are highly correlated with the exact probabilities, which should follow from having access to the exact normalization constant. Mainly three observations can be made in this plot. The first observation is that LBP with exact normalization constant has a larger tendency of overestimating probabilities close to zero than underestimating them. This could be related to similar behavior we saw for MH-LBP. The second observation is the strong trend that LBP with exact normalization constant consistently underestimates higher probabilities, and almost never underestimates it. This seems like an extreme case of what we saw for MH-LBP, and begs the question whether this might be a trend for such approximate schemes, or at least methods like LBP. Lastly, there seems to be an almost linearly increasing tendency to underestimate increasing probabilities, which we see from the widening point cloud above the correlation line.

6.5 Failed convergence of MH-LBP

In exactly *one* case out of in total 111887 clusters the MH-LBP algorithm failed to converge to a solution, and instead the messages in the graph oscillates between two values. These two values are not necessarily the same for each message. A software bug caused many more clusters to not converge, about 780, and so at the point of writing it is unclear whether the failed convergence is due to another software bug or whether this is a more fundamental fact in the way the factor graph is structured and the messages are defined. A small discussion of this particular case is anyhow warranted.

They prove mathematically in [13] that the track-to-measurement and measurement-to-track messages, $\mu_{t\to j}$ and $\nu_{j\to t}$, respectively, must converge, and this was assumed to hold for the message definitions in (4.17) and (4.18). Namely, if we were to fix σ_t for all t, then we expect $\mu_{t\to j}$ and $\nu_{j\to t}$ to converge. We therefore conclude that the main culprit for the oscillations are the σ_t messages. In [16] they consider a similar association problem, only instead of multihypothesis it is multiscan, and state that the Bethe free energy for this association graph is nonconvex, which results in undesirable behavior. A possible explanation for the nonconvergence could be that the Bethe free energy function of the multihypothesis association graph is similarly nonconvex or exhibits other undesirable properties.

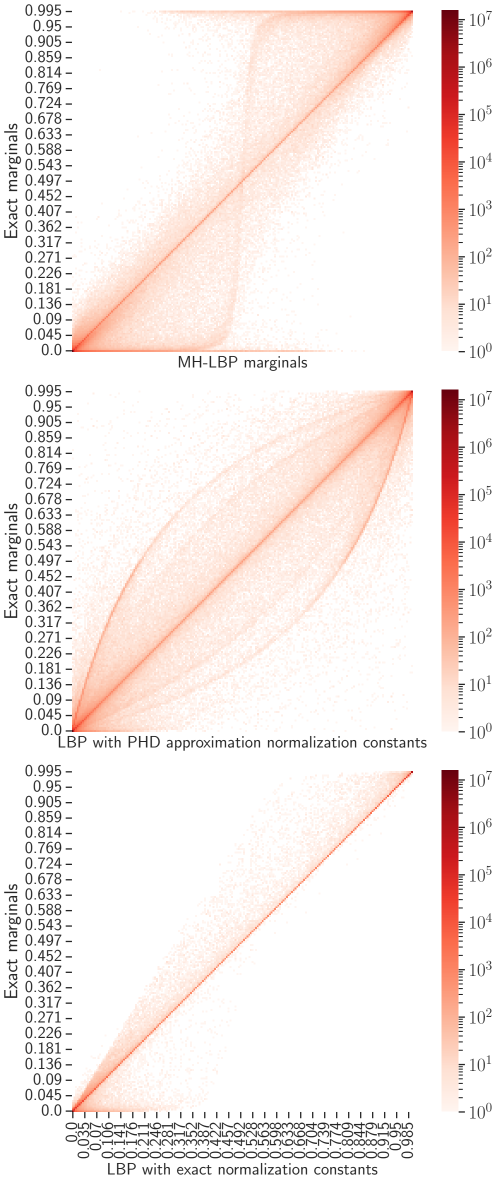


Figure 6.4: *Heatmap showing correlation between MH-LBP marginals and exact marginals. Note that the colors are logarithmic.*

The oscillations of the hypothesis-to-track messages, σ_t, can be found in Figure 6.5, together with the *reward matrix* for the cluster, which is here with one row for each track and one column for each measurement and one for misdetection. The values placed in each entry in the matrix depends on the index of the entry, where the entries at $(t, 1)$ contains the misdetection probability of each track and (t, j'), $j' > 2$ contains the likelihood for the association between track t and measurement j, $j = j' - 1$. In Figure 6.5 we see that track 9 and 11 have the largest oscillations, while also being the tracks with the largest likelihoods in the reward matrix.

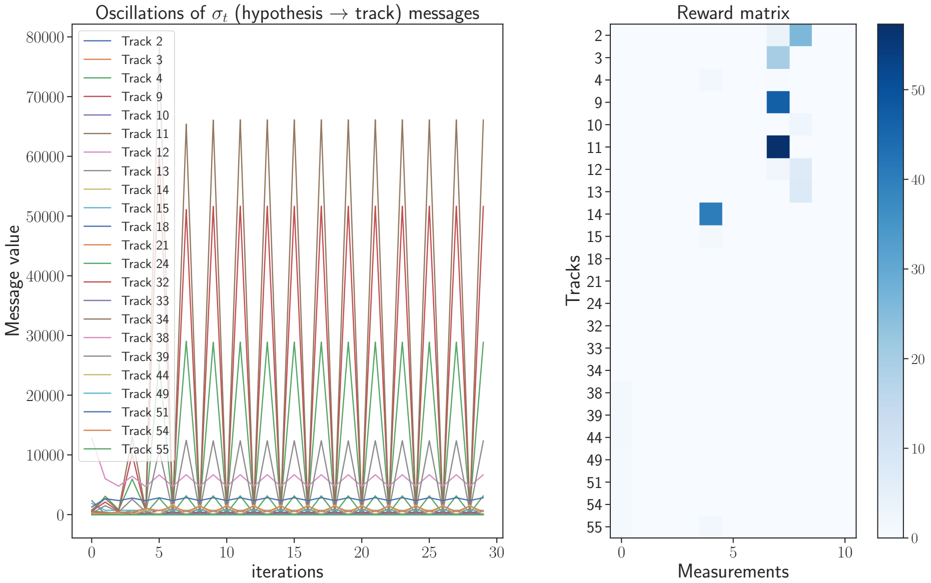


Figure 6.5: Oscillations of the different sigma messages of the tracks in the cluster where MH-LBP did not converge, together with the cluster-conditioned reward matrix. Note that only 30 *iterations are plotted.*

In [13] results show that the accuracy of LBP is tied to the *Signal-to-noise-ratio* (SNR) of the problem, where lower SNR seems to improve accuracy in vice versa. In other words, for high misdetection probabilities and clutter rate, we can expect LBP in the hypothesis-conditioned case to have improved accuracy. In [26] they observe that priors in a graph with low values can cause oscillations in LBP, and that increasing these in their experiments helped with convergence. As the misdetection probablities do appear in the priors of the tracks in our factor graph, it was tested with considerable higher misdetection probabilities. In Figure 6.6 we see that after adding 2.5 to the log misdetections we achieve convergence, although relatively slowly as the σ_t messages oscillates but the amplitude slowly decays. The fact that the messages converges for low SNR might be related to the discussion in Chapter 5.1.1, as we expect for large detection likelihoods that we get stronger correlation in the association graph, and that making the misdetection probability larger makes these likelihood relatively smaller, weakening the

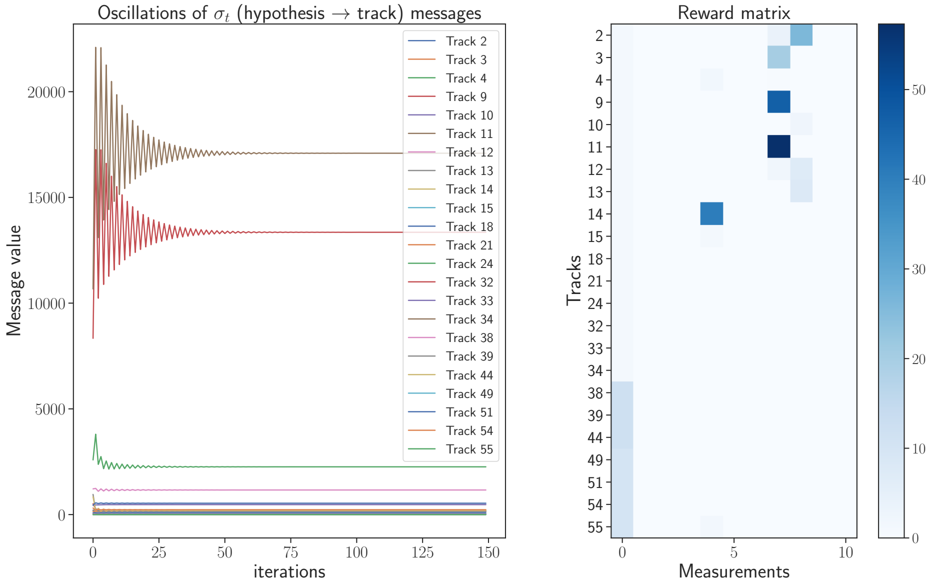


Figure 6.6: *Convergence of messages after increasing the misdetection probabilities. The logarithm of the misdetection probabilities were increased by* 2.5*, effectively multiplying them by* $e^{2.5}$*. Note that* 150 *iterations are plotted to capture the convergence.*

correlation. Lastly, we also see that, comparing Figure 6.5 and Figure 6.6, it seems like the size of the likelihoods are related to how large the σ_t messages are, as track 9 and 11 dominate in both cases, but not to its convergence properties.

IV

Closing remarks

7 | Conclusion

This report started by introducing the challenges of data association in multi-target tracking. The challenges are attributed to the combinatorial complexity that follows from allowing multiple tracks to compete for the same measurements, and reasons that approximations are necessary in practice. Based on previous work in data association in MTT that are LBP, the main contribution of this paper is to generalize the approximate methods for a multihypothesis setting.

Prior to presenting the novel methods for multihypothesis data association in multi-target tracking, factor graphs are introduced, together with a presentation of the inference methods BP for tree-structured graphs and LBP for general graphs contain loops. Following that, data association in MTT is reviewed, where the modelling assumptions necessary for deriving a joint association hypothesis posterior in MTT are established. These are then used to derive said posterior distribution in the MTT trackers JPDA and MHT, together with how the posterior in MHT can be modified to become equal to the one used in PMBM. Finally, a description for how to do exact marginalization of the joint posterior is presented, together with a common approach for approximating the marginals by enumerating only the M most likely association hypotheses with Murty's method.

After the preliminaries, the main contributions of this report are presented. The first novelty is the factor graph representation of the joint multihypothesis association hypothesis posterior, together with the introduction of the nonexistence state of tracks which allows for efficient recycling of tracks in PMBM. The, two novel algorithms for approximate marginalization of the joint multihypothesis association hypothesis posterior are presented. The first method, called MH-LBP, does efficient LBP on the full posterior by deriving special messages that use clever normalization to reduce computation and memory complexity. The second method, called LBP-PHD, does hypothesis-conditioned LBP over all prior hypotheses by also estimating the associated hypothesis-conditioned likelihood with a Poisson approximation inspired by the PHD filter.

The proposed methods are tested on a simulated dataset where the data is generated by a PMBM filter implementation, and three methods are compared. The first two methods are the novel methods MH-LBP and LBP-PHD. The third method is a best-case comparision that does hypothesis-conditioned LBP in the same manner as LBP-PHD, but with exact hypothesis-conditioned likelihood. The results show that both MH-LBP and LBP-PHD perform well in most cases. The largest differences between was attributed to the Poisson approximation of the hypothesis-conditioned likelihood, as it showed a tendency to overestimate low values of the true likelihood and underestimate high values. Based on the correlation between the estimated marginals and the exact marginals, MH-LBP seems to overall estimate more conservative marginals, while LBP-PHD shows larger variance. Inspecting the performance of the best-case hypothesis-conditioned LBP shows promise in computing the association marginals by hypothesis-conditioned LBP given an accurate estimate of the corresponding likelihood can be found, in particular for track recycling, as it is able to estimate the nonexistence probability with very high accuracy. A single case of nonconverging MH-LBP is discussed. It is speculated that this is due to oscillations in the hypothesis-to-track messages and that this is related to the SNR of the association case.

7.1 Future work

The most promising path forward is to further investigate the possibilities in variational inference, where mainly two approaches can be identified.

The first is to improve MH-LBP by iterating and improving on the LBP scheme. This is especially interesting considering the nonconvergence case of MH-LBP discussed in Chapter 6.5. Choosing the Bethe approximation is only one choice to make in the optimization problem discussed in Chapter 2.4.1. A common choice is the even simpler trial distribution q that factorizes as a product marginals, i.e. where all variables are assumed independent, which is called the *mean field approximation* [54]. Another option is to work directly on the Bethe free energy and modify it for better properties when optimizing. This is done by Williams et. al do in [16] that was briefly mentioned in Chapter 6.5. They recognize that the underlying Bethe free energy that LBP optimizes is nonconvex for the multiscan data association graph they construct, and addresses this by a *convexification* of the Bethe free energy by using*fractional free energy* [55], [56] together with deriving a BP-like scheme for approximating marginals. A similar inspection and approach might be necessary for the multihypothesis data association graph presented in this work.

The second approach is to turn to the LBP-PHD method and use the fact that optimizing the variational free energy functional does not only compute approximate marginals, but also allow for computing an approximation of the normalization constant of the true distribution. As proven in [13], LBP on the hypothesis-condtioned data association problem has many desirable properties, the most important perhaps being guaranteed convergence. This makes approximating the hypothesis-conditioned likelihood a favorable alternative if an accurate estimate can be computed. Although the PHD approximation in the LBP-PHD method showed promise, comparing its performance to the best-case with exact likelihood showed that there is much room for improvement. In [15], Vontobel approximates the *permanent* of a matrix by using LBP in the same way as Williams et. al does in [13], only he uses the fixed-point messages to compute the approximate Bethe normalization constant. This approach seems promising for a more accurate estimate of the hypothesis-conditioned likelihood.

V

Appendices

Bibliography

[1] V. Isham, 'An introduction to spatial point processes and markov random fields,' *International Statistical Review / Revue Internationale de Statistique*, vol. 49, no. 1, p. 21, 1981. doi: 10.2307/1403035.

[2] R. Kindermann and J. L. Snell, 'Markov random fields and their applications,' *Contemporary Mathematics*, 1980. doi: 10.1090/conm/001.

[3] C. J. Preston, *Gibbs states on countable sets*. Cambridge Univ. Press, 1974.

[4] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[5] F. V. Jensen, *An introduction to bayesian networks*. Springer, 1996.

[6] R. E. Kalman, 'A new approach to linear filtering and prediction problems,' *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[7] Kschischang, Frank R. and Frey, Brendan J. and Loeliger, Hans-Andrea, 'Factor Graphs and the Sum-Product Algorithm,' *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[8] L. Chen, M. Wainwright, M. Cetin and A. Willsky, 'Multitarget-multisensor data association using the tree-reweighted max-product algorithm,' *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5096, May 2003. doi: 10.1117/12.496939.

[9] L. Chen, M. Cetin and A. Willsky, 'Distributed data association for multi-target tracking in sensor networks,' *Information Fusion - INFFUS*, Jan. 2005.

[10] L. Chen, M. J. Wainwright, M. cCetin and A. S. Willsky, 'Data association based on optimization in graphical models with application to sensor networks,' *Mathematical and Computer Modelling*, vol. 43, no. 9, pp. 1114–1135, 2006, Optimization and Control for Military Applications, issn: 0895-7177. doi: https:

//doi.org/10.1016/j.mcm.2005.12.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717705005406.

[11] M. Cetin, L. Chen, J. Fisher *et al.*, 'Distributed fusion in sensor networks,' *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 42–55, 2006. DOI: 10.1109/MSP.2006.1657816.

[12] J. L. Williams and R. A. Lau, 'Data association by loopy belief propagation,' in *2010 13th International Conference on Information Fusion*, 2010, pp. 1–8. DOI: 10.1109/ICIF.2010.5711833.

[13] J. L. Williams and R. A. Lau, 'Approximate evaluation of marginal association probabilities with belief propagation,' *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2942–2959, Oct. 2014, arXiv:1209.6299 [cs], ISSN: 0018-9251, 1557-9603, 2371-9877. DOI: 10.1109/TAES.2014.120568. [Online]. Available: http://arxiv.org/abs/1209.6299 (visited on 30th Aug. 2022).

[14] J. L. Williams, 'Marginal multi-bernoulli filters: Rfs derivation of mht, jipda, and association-based member,' *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, 1664–1687, 2015. DOI: 10.1109/taes.2015.130550.

[15] P. O. Vontobel, 'The bethe permanent of a nonnegative matrix,' *IEEE Transactions on Information Theory*, vol. 59, no. 3, 1866–1901, 2013. DOI: 10.1109/tit.2012.2227109.

[16] J. L. Williams and R. A. Lau, 'Multiple scan data association by convex variational inference,' *IEEE Transactions on Signal Processing*, vol. 66, no. 8, pp. 2112–2127, 2018. DOI: 10.1109/TSP.2018.2802460.

[17] F. Meyer, P. Braca, P. Willett and F. Hlawatsch, 'Scalable multitarget tracking using multiple sensors: A belief propagation approach,' in *2015 18th International Conference on Information Fusion (Fusion)*, 2015, pp. 1778–1785.

[18] F. Meyer, P. Braca, F. Hlawatsch, M. Micheli and K. D. LePage, 'Scalable adaptive multitarget tracking using multiple sensors,' in *2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6. DOI: 10.1109/GLOCOMW.2016.7849034.

[19] F. Meyer, P. Braca, P. Willett and F. Hlawatsch, 'A scalable algorithm for tracking an unknown number of targets using multiple sensors,' *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3478–3493, 2017. DOI: 10.1109/TSP.2017.2688966.

[20] D. Gaglione, G. Soldi, F. Meyer *et al.*, 'Bayesian information fusion and multitarget tracking for maritime situational awareness,' *IET Radar, Sonar; Navigation*, vol. 14, no. 12, 1845–1857, 2020. DOI: 10.1049/iet-rsn.2019.0508.

[21] D. Gaglione, P. Braca and G. Soldi, 'Belief propagation based ais/radar data fusion for multi - target tracking,' in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 2143–2150. DOI: 10.23919/ICIF.2018.8455217.

[22] J. L. Williams, 'Graphical model approximations to the full bayes random finite set filter,' *CoRR*, vol. abs/1105.3298, 2011. arXiv: 1105.3298. [Online]. Available: http://arxiv.org/abs/1105.3298.

[23] L. Zhu, J. Wang and S. Yang, 'Factor graphs based iterative decoding of turbo codes,' *IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions*, DOI: 10.1109/icccas.2002.1180569.

[24] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques* (Adaptive computation and machine learning). MIT Press, 2009, ISBN: 9780262013192. [Online]. Available: https://books.google.co.in/books?id=7dzpHCHzNQ4C.

[25] C. M. BISHOP, *Pattern recognition and machine learning*. SPRINGER-VERLAG NEW YORK, 2016.

[26] K. P. Murphy, Y. Weiss and M. I. Jordan, 'Loopy belief propagation for approximate inference: An empirical study,' *CoRR*, vol. abs/1301.6725, 2013. arXiv: 1301.6725. [Online]. Available: http://arxiv.org/abs/1301.6725.

[27] R. McEliece, D. MacKay and J.-F. Cheng, 'Turbo decoding as an instance of pearl's "belief propagation" algorithm,' *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, 140–152, 1998. DOI: 10.1109/49.661103.

[28] J. S. Yedidia, W. Freeman and Y. Weiss, 'Generalized belief propagation,' in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich and V. Tresp, Eds., vol. 13, MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper/2000/file/61b1fb3f59e28c67f3925f3c79be81a1-Paper.pdf.

[29] J. Yedidia, W. Freeman and Y. Weiss, 'Constructing free-energy approximations and generalized belief propagation algorithms,' *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, 2005. DOI: 10.1109/TIT.2005.850085.

[30] 'Statistical theory of superlattices,' *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, vol. 150, no. 871, 552–575, 1935. DOI: 10.1098/rspa.1935.0122.

[31] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[32] A. T. Ihler, J. W. F. III and A. S. Willsky, 'Loopy belief propagation: Convergence and effects of message errors,' *Journal of Machine Learning Research*, vol. 6, no. 31, pp. 905–936, 2005. [Online]. Available: http://jmlr.org/papers/v6/ihler05a.html.

[33] E. Brekke, 'Fundamentals of Sensor Fusion,' en, p. 320,

[34] T. I. Fossen, *Handbook of Marine Mraft Hydrodynamics and Motion Control*. Wiley, 2014.

[35] P. Y. C. H. Robert Grover Brown, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, 4th ed. Wiley, 2012, ISBN: 0470609699,9780470609699.

[36] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, 1st ed. The MIT Press, 2005, ISBN: 0262201623,9780262201629.

[37] T. 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, Jul. 1983, Conference Name: IEEE Journal of Oceanic Engineering, ISSN: 1558-1691. DOI: 10.1109/JOE.1983.1145560.

[38] K. Granström and M. Baum, 'Extended object tracking: Introduction, overview and applications,' *CoRR*, vol. abs/1604.00970, 2016. arXiv: 1604.00970. [Online]. Available: http://arxiv.org/abs/1604.00970.

[39] Bar-Shalom, Y. and Tse, E., 'Tracking in a cluttered environment with probabilistic data association,' *Automatica*, vol. 11, no. 5, 451–460, Sep. 1975. DOI: 10.1016/0005-1098(75)90021-7.

[40] D. Reid, 'An algorithm for tracking multiple targets,' *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979, Conference Name: IEEE Transactions on Automatic Control, ISSN: 1558-2523. DOI: 10.1109/TAC.1979.1102177.

[41] D. Musicki, R. Evans and S. Stankovic, 'Integrated probabilistic data association,' *IEEE Transactions on Automatic Control*, vol. 39, no. 6, 1237–1241, 1994. DOI: 10.1109/9.293185.

[42] D. Musicki and R. Evans, 'Joint integrated probabilistic data association - jipda,' *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, DOI: `10.1109/icif.2002.1020938`.

[43] Y. Bar-Shalom, S. S. Blackman and R. J. Fitzgerald, 'Dimensionless score function for multiple hypothesis tracking,' *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 392–400, 2007. DOI: `10.1109/TAES.2007.357141`.

[44] E. F. Brekke, A. G. Hem and L.-C. N. Tokle, 'The vimmjipda: Hybrid state formulation and verification on maritime radar benchmark data,' in *Global Oceans 2020: Singapore – U.S. Gulf Coast*, 2020, pp. 1–5. DOI: `10.1109/IEEECONF38699.2020.9389007`.

[45] I. Cox, M. Miller, R. Danchick and G. Newnam, 'A comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence,' *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 295–301, 1997. DOI: `10.1109/7.570789`.

[46] 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, ISSN: 0030364X, 15265463. [Online]. Available: `http://www.jstor.org/stable/168595` (visited on 18th Dec. 2022).

[47] M. Miller, H. Stone and I. Cox, 'Optimizing murty's ranked assignment method,' *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 851–862, 1997. DOI: `10.1109/7.599256`.

[48] R. Danchick and G. Newnam, 'Reformulating reid's mht method with generalised murty k-best ranked linear assignment algorithm,' *IEE Proceedings - Radar, Sonar and Navigation*, vol. 153, no. 1, p. 13, 2006. DOI: `10.1049/ip-rsn:20050041`.

[49] E. F. Brekke and L.-C. N. Tokle, 'Hypothesis exploration in multiple hypothesis tracking with multiple clusters,' *2022 25th International Conference on Information Fusion (FUSION)*, 2022. DOI: `10.23919/fusion49751.2022.9841311`.

[50] H. W. Kuhn, 'The hungarian method for the assignment problem,' *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, 83–97, 1955. DOI: `10.1002/nav.3800020109`.

[51] S. S. Blackman and R. Popoli, *Design and analysis of Modern Tracking Systems*. Artech House, 1999.

[52] R. Jonker and A. Volgenant, ‘A shortest augmenting path algorithm for dense and sparse linear assignment problems,’ *Computing*, vol. 38, no. 4, 325–340, 1987. DOI: 10.1007/bf02278710.

[53] M. R. P. S., *Statistical multisource-multitarget information fusion*. Artech House, 2007.

[54] T. S. Jaakkola, ‘Tutorial on variational approximation methods,’ *Advanced Mean Field Methods*, 2001. DOI: 10.7551/mitpress/1100.003.0014.

[55] A. B. Yedidia and M. Chertkov, ‘Computing the permanent with belief propagation,’ *CoRR*, vol. abs/1108.0065, 2011. arXiv: 1108.0065. [Online]. Available: http://arxiv.org/abs/1108.0065.

[56] J. L. Williams, ‘Interior point solution of fractional bethe permanent,’ in *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, 2014, pp. 213–216. DOI: 10.1109/SSP.2014.6884613.