Abstract

Recent results in single-hypothesis, multi-target tracking uses loopy belief propagation (LBP) to perform efficient, approximate marginalization of the association hypothesis posterior with much success. In this work, we generalize this methodology to a multi-cluster, multi-hypothesis setting by presenting four novel methods. The methods are designed to marginalize and estimate the normalization constant of the novel multi-cluster, multi-hypothesis association graph presented in this work. The normalization constants are estimated with novel, specialized expressions for computing the Bethe constant of the association graph.

All presented methods are based on LBP. One method uses specialized LBP messages that are optimized for efficient computation and memory management on the full association graph. The three other methods are based on a novel cluster-conditioning method also presented in this work that avoids reenumerating the prior hypotheses in cases where clusters merge by having a marginalization step that reintroduces independence between the clusters. One of these methods uses single-cluster, multi-hypothesis LBP to do inference on the single-cluster, multi-hypothesis association graph. The two other methods marginalizes over the prior hypotheses to do inference on single-cluster, single-hypothesis association graphs.

The motivation for developing these presented methods is to efficiently do marginalization of the multi-cluster, multi-hypothesis association posterior to enable track recycling in the Poisson Multi-Bernoulli Mixture (PMBM) and improve filter consistency. Results show that basing the normalization constant estimate on the Bethe constant shows promise and that hypothesis-conditioned LBP gives more accurate and reliable marginal estimates than multi-hypothesis LBP.

Sammendrag

Nyere resultater innenfor målfølging av flere mål med kun én hypotese bruker "loopy belief propagation" (LBP) for å utføre effektiv, tilnærmet marginalisering av hypotese-posterioren med stor suksess. Denne avhandlingen generaliserer denne metoden til det generelle målfølgingsscenarioet med flere samlinger av mål og flere hypoteser ved å presentere fire nye metoder. Disse metodene er designet for å marginalisere og estimere normaliseringskonstanten til den nye assosiasjonsgrafen for flere samlinger av mål og flere hypoteser som blir presentert i dette arbeidet. Normaliseringskonstantene blir estimert ved hjelp av nye, spesialiserte uttrykk for beregning av Bethe-konstanten til assosiasjonsgrafen.

En metode bruker spesialiserte LBP-meldingsdefinisjoner som er optimalisert for effektiv beregning og minnehåndtering på den fulle assosiasjonsgrafen. De tre andre metodene er basert på en ny betingelsesmetode over samlingene av mål som også blir presentert i dette arbeidet. Denne metoden unngår å enumerere hypoteser på nytt i tilfelle samlinger slås sammen ved å ha et marginaliseringstrinn som gjeninnfører uavhengighet mellom samlingene. En av disse metodene bruker LBP på en assosiasjonsgraf for en enkel samling av mål over flere hypoteser for inferens. De to andre metodene tar marginalisering videre ved å marginalisere over hypotesene i målsamlingen for å gjøre inferens på assosiasjonsgrafter for en enkel samling med en hypotese.

Motivasjonen for å utvikle disse metodene er å muliggjøre såkalt resirkulering av målestimat i Poisson Multi-Bernoulli-mikstur (PMBM) filteret for å forbedre ytelsen til filteret ved å effektivt approksimere assosiasjonsmarginaler. Resultatene viser at normaliseringskonstantestimat basert på Bethe-konstanten virker lovende, og at hypotesebetinget LBP gir mer nøyaktige og pålitelige marginalestimater enn LBP over flere hypoteser.

Preface

The present work is the resulting master thesis in TTK4900 at the department of Engineering Cybernetics spring 2023.

I would like to thank my supervisor Edmund Førland Brekke and co-supervisor Lars-Christian Ness Tokle. I will forever be grateful for how incredibly available you have been for technical discussions. Considering I wrote this thesis alone and therefore had no one else to confer with, you have both been invaluable to the result. It has truly been an honor pursuing a new state-of-the-art with you.

I would also like to thank Zeabuz AS and my co-supervisor there, Erik Falmår Wilthil. Granting me a part-time job where I could use my competence in target tracking has been an incredibly fun and rewarding experience. I look forward to start working full-time with you this fall.

This thesis concludes an incredible eight year long study period. Therefore, I would like to thank my parents, Anita and Geir, for supporting me and encouraging me all these years. I will never doubt postponing graduation to be able to do everything I wanted as I student.

Odin Aleksander Severinsen

Trondheim, June 2023

Table of Contents

V Appendices

List of Abbreviations

BP Belief propagation

FISST Finite set statistics

JPDA Joint probabilistic data association

LBP Loopy belief propagation

MB Multi-Bernoulli
MBM Multi-Bernoulli Mixture
MCMH-LBP Multi-cluster, multi-hypothesis loopy belief propagation
MH-LBP Multi-hypothesis loopy belief propagation
MHT Multiple hypothesis tracker
MRF Markov Random Field
MTT Multi-target tracking

PGF Probability generating function
PGFL Probability generating functional
PHD Probability hypothesis density
PMB Poisson multi-Bernoulli
PMBM Poisson multi-Bernoulli mixture

RFS Random finite set

SNR Signal-to-noise ratio

I

Introduction and preliminaries

1 | Introduction

Situational awareness is the ability for an autonomous system to understand its environment and enables it to operate truly autonomously by adapting to it. An example of this in practice can be an autonomous ferry that shall cross a canal and needs to plan a route that does not interfere with traffic. In this thesis we concern ourselves with *target tracking* which involves detecting, estimating, and predicting the state of external *targets*, which in general can be any object existing in the real world, both static and dynamic. In practice we do this by constructing *tracks*, which are estimated trajectories for the targets we track.

In this thesis we will focus on *Multi-target tracking* (MTT), the ability to track multiple, interacting tracks simultaneously. A core part of any such system is *data association*. In broad terms, this is the logic for determining whether detections received from some exteroceptive sensor are of actual tracks, in which case *which tracks* if ambiguous, or if some or all detections are *false alarms*, meaning they should be discarded. Even with severe restrictions and assumptions, solving this problem exactly is in practice intractable. This follows from the combinatorial complexity of the problem – to do exact data association we have to enumerate all such ways of explaining the detections, called *association hypotheses*. Without any pruning techniques, the number of association hypotheses that can be made from the detections received over multiple timesteps grows astronomically large. Therefore, approximations have to be made, and exploring such approximations is the core content of this thesis.

In particular, the following thesis explores novel, approximate methods specialized for MTT that utilize a specific structure in the data association problem. The structure together with approximation techniques are what enables efficient data association to perform MTT for online purposes, which is the main, overall goal of the presented work.

1.1 Related work

This thesis is a continuation of the work done in a pre-master project in 2022 that ended in a project report. The methods and results presented in this report has been accepted for publication in the proceedings of the 2023 26th International Conference on Information Fusion (FUSION) [1]. A copy of the accepted paper can also be found in Appendix C.

The pre-master project focused on a particular form of MTT called *multi-hypothesis* MTT, where we keep the association hypotheses that are made in a timestep and uses them to form the association hypotheses of the subsequent timesteps. This is contrary to *single-hypothesis* MTT where approximations are used to combine the association hypotheses into a single hypothesis at the end of each timestep.

There were three novel contributions presented in the pre-master project. The first was a novel, *graphical representation* of the multi-hypothesis, single-cluster association posterior that appears in multi-hypothesis MTT. Graphical models for probabilistic inference has been widely used in robotics and intelligent systems for several decades. Graphical modelling varies based on representation abilities of the underlying probability density, and the first representations that saw use are *Markov random fields* [2]–[4] and *Bayesian networks* [5], [6]. The pre-master project used a graphical representation called *factor graph*, a representation first presented by Kschischang et. al in [7]. The second novelty was a specialized formulation of *Loopy belief propagation* (LBP), called *Multi-hypothesis loopy belief propagation* (MH-LBP), to perform full LBP directly on the multi-hypothesis association to retrieve approximate association marginals from the original, joint association posterior. The LBP method was first described by Pearl in [5] who formulated the method for probabilistic graphs with *tree structures*, where the method is exact. The LBP approach was a generalization based upon the work of Williams et. al in [8], which uses a similar approach to approximate the association marginals in a single-hypothesis context. The last novelty was an alternative marginalization approach that performed LBP on hypothesis-conditioned association posteriors, where the association graph was identical to the one in [8]. The approximated marginals were then combined by total probability using suitable scaling from an approximated hypothesis-conditioned likelihood. In the pre-master project, the approximation was calculated using ideas from the *Probability hypothesis density* (PHD) filter [9] which is based upon *Finite set statistics* (FISST) [9].

Factor graphs for inference in multi-target tracking has seen other use as well. The initial usage of factor graphs was pioneered by Chen, Cetin et al [10]–[13]. In their work, they employed the *max-product* algorithm, a variant of LBP, to determine the optimal

association hypothesis by maximizing a joint distribution instead of marginalizing it.

Meyer, Braca et al. integrated the data association method proposed in [8] into a factor graph representation of the joint track state posterior for multi-sensor scenarios. They utilized LBP to approximate the marginal track state posteriors. Furthermore, they extended their approach to include estimation of time-varying model parameters in [14] and later handle an unknown number of targets in [15].

Finally, Gaglione et al. presented a method for multi-sensor, multi-target tracking in the maritime domain, utilizing a specifically designed factor graph and employing LBP for approximate inference [16]. Additionally, in [17], the same authors used LBP in order to fuse radar and Automatic Identification System (AIS) data.

1.2 Problem description and main contributions

In the pre-master project, the main contribution was on approximate marginalization of the multi-hypothesis association posterior. This was driven by the observation that data association, and in particular marginalization, in previous works was primarily done in a single-hypothesis setting. The main motivation for generalizing to multi-hypothesis was to allow for better *track management* in multi-hypothesis tracking to achieve more consistent and robust tracking. The pre-master project, however, restricted itself to inference on a *single-cluster* scenario, i.e., where we only consider a single collection of tracks that interact with each other.

In this thesis we first and foremost generalize this work for *multi-cluster* purposes, i.e. where we have multiple, independent clusters of tracks at the same time. We motivate this for the following reason. In a practical implementation of an MTT filter we might want to regulate the number of hypotheses we keep. Mainly this is done for computational reasons, but additionally we do not always require a large set of association hypotheses, or even more than one, to maintain estimation accuracy. We do, however, in general *always track multiple clusters of tracks*, and so inference for single-cluster tracking is not particularly useful in practice. Generalizing for multi-cluster therefore allows for using the inference methods in a practical implementation, which is the end goal.

In particular, a considerable challenge in multi-hypothesis, multi-cluster MTT is *cluster merging*, which is when clusters, i.e. independent collections of tracks, interact with each other. In general, this increases the number of hypotheses in the resulting cluster enormously. Although efficient implementations exist that can enumerate posterior hypotheses in the update step [18], they prune a large proportion of the hypothesis space, discarding information. The methods presented in this work are formulated in such a way

as to avoid hypothesis enumeration. This is intended to keep more information intact.

The following thesis presents four novelties, all in Part II:

- A factor graph representation of the multi-cluster, multi-hypothesis association posterior, which is presented in Chapter 5. This novel formulation is necessary as it lays the foundation for the presented methods.
- A generalization of the efficient LBP scheme from the preceding project report for the multi-cluster scenario, called *Multi-cluster, multi-hypothesis loopy belief propagation* (MCMH-LBP), which is presented in Chapter 6. The LBP method has multiple times previously been seen to give good approximations in a large number of cases for low computational costs, a property that is particularly desirable in an MTT pipeline.
- In the same chapter we also present novel expressions for estimating the *normalization constant* to the multi-cluster, multi-hypthesis association posterior, single-cluster, multi-hypothesis association posterior and lastly, single-cluster, single-hypothesis association posterior based on the *Bethe approximation* of the respective distributions using the corresponding, specialized LBP messages. We argue why and how to use these estimates to improve filter consistency and robustness, in particular when doing multi-hypothesis tracking.
- A completely novel overparameterization of the factor graph presented in Chapter 5 supplemented by a novel, flexible multi-cluster, multi-hypothesis marginalization procedure adaptable to any exact or approximate marginalization scheme intended for single-cluster and single- or multi-hypothesis inference, is presented in Chapter 7. Based on this framework we additionally describe in detail how to use it for more efficient, exact multi-cluster, multi-hypothesis marginalization using existing single-cluster, single-hypothesis solvers and also propose three different variations for approximate inference using LBP.

1.3 Outline

The thesis is structured as follows. First, the theory of *factor graphs* and LBP is introduced in Chapter 2 that forms the fundament for approximate inference in the present work. In Chapter 3, a review of MTT concepts and modeling assumptions is provided, largely based on the same review in the preceding project report. Then, in Chapter 4, we formally introduce the *Poisson multi-Bernoulli mixture* (PMBM) filter in its most general form. We also state the involved filtering equations used to later be able to build the required

joint association posterior for data association. The main chapters of this thesis are Chapters 5 to 7 where the novelties are presented. Then, in Chapter 8 we inspect how MCMH-LBP in particular performs on a simple test case where the system parameters are tweaked to try to better understand the dynamics of LBP and how the marginal and normalization constant estimates are affected by different parameters. Chapter 9 introduces the large-scale, simulated dataset that the proposed methods are tested on, with results and discussion in Chapter 10. Lastly, a conclusion with future work can be found in Chapter 11.

2 | Probabilistic inference in graphical models

At the very core of the data association methods in the present thesis are *graphical representations of probability distributions* and *inference algorithms* applied to such models. The main benefit of doing this is to better encode and, crucially, exploit structure between the variables of the underlying distribution. The following chapter will present the main graphical representation used, the *factor graph*, and also briefly mention an alternative representation called the *Markov random field*. Lastly, a central, approximate inference algorithm, *loopy belief propagation*, is presented and discussed.

2.1 Probability distributions

In general, a probability distribution is either a *probability density* or a *probability mass function*, and both are relevant for the present work.

Without loss of generality, a probability density is a function $p(x_1, \ldots, x_n)$ over the real, continuous, stochastic variables $x_i \in \mathbb{R}$, $i \in \{1, \ldots, n\}$ that satisfies

$$p(x_1, \ldots, x_n) \geq 0, \forall x_1, \ldots, x_n, \tag{2.1}$$

$$\int \cdots \int p(x_1, \ldots, x_n) \, \mathrm{d}x_1 \ldots \mathrm{d}x_n = 1. \tag{2.2}$$

Note that a probability density does not evaluate to a probability, but must instead be integrated first.

Again, without loss of generality, a probability mass function $\mathsf{Pr}\{x_1, \ldots, x_n\}$ is the discrete counterpart for discrete variables over some *alphabet* $\mathbb{X}$, $x_i \in \mathbb{X}$, $i \in \{1, \ldots, n\}$.

Specifically, we will focus on *categorical* variables where the elements of $\mathbb{X}$ carry a semantic meaning and are not intrinsically ordered. Contrary to probability densities, a

probability mass function does evaluate to an actual probability. Additionally, it satisfies

$$0 \leq \mathsf{Pr}\{x_1, \ldots, x_n\} \leq 1, \forall x_1, \ldots, x_n, \tag{2.3}$$

$$\sum_{x_1,\ldots,x_n} \mathsf{Pr}\{x_1, \ldots, x_n\} = 1, \tag{2.4}$$

where $\sum_{x_1,\ldots,x_n}$ is short-hand notation for the iterated sum $\sum_{x_1 \in \mathbb{X}} \cdots \sum_{x_n \in \mathbb{X}}$.

2.1.1 Exploiting conditional independence

For the sake of argument, assume we have a system of latent, discrete variables $x_1, \ldots, x_n$. When doing inference on such a system, we typically model the joint distribution $\mathsf{Pr}\{x_1, \ldots, x_n\}$ and either try to maximize this function or marginalize it to independent marginal distributions $\mathsf{Pr}\{x_i\}$ for each x_i. If we focus on the latter objective, the procedure is in principle a matter of summing out the other variables, i.e.,

$$\mathsf{Pr}\{x_i\} = \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_n} \mathsf{Pr}\{x_1, \ldots, x_n\}\,. \tag{2.5}$$

Unfortunately, performing the computation in (2.5) becomes prohibitively expensive when the number of variables and the size of the alphabet $\mathbb{X}$ grows arbitrarily large. This follows from the fact that the joint distribution is factorized according to the *chain rule of probability*,

$$\mathsf{Pr}\{x_1, \ldots, x_n\} = \mathsf{Pr}\{x_i\}\,\mathsf{Pr}\{x_1 \mid x_i\} \cdot \cdots \cdot \mathsf{Pr}\{x_n \mid x_1, \ldots, x_{i-1}, x_{i-1}, \ldots, x_{n-1}\}\,. \tag{2.6}$$

Assuming we form $\mathsf{Pr}\{x_i\}$ as a table of probabilities, the iterated sum must be computed for each element of $\mathbb{X}$, i.e. $|\mathbb{X}|$ times, in addition to, for each time we compute the iterated sum, at most having to do a sum over the table $\mathsf{Pr}\{x_n \mid x_1, \ldots, x_{i-1}, x_{i-1}, \ldots, x_{n-1}\}$ which is $\mathcal{O}(|\mathbb{X}|^{n-1})$ large, i.e., exponential in number of variables.

If now instead assume that e.g. we can write (2.6) as

$$\begin{aligned}
\mathsf{Pr}\{x_1, \ldots, x_n\} &= \mathsf{Pr}\{x_i\}\,\mathsf{Pr}\{x_1 \mid x_i\} \cdot \cdots \cdot \mathsf{Pr}\{x_n \mid x_1, \ldots, x_{i-1}, x_{i-1}, \ldots, x_{n-1}\} \\
&= \mathsf{Pr}\{x_i\} \\
&\cdot \mathsf{Pr}\{x_{i+1} \mid x_i\}\,\mathsf{Pr}\{x_{i+2} \mid x_{i+1}\} \cdot \cdots \cdot \mathsf{Pr}\{x_n \mid x_{n-1}\} \\
&\cdot \mathsf{Pr}\{x_{i-1} \mid x_i\}\,\mathsf{Pr}\{x_{i-2} \mid x_{i-1}\} \cdot \cdots \cdot \mathsf{Pr}\{x_1 \mid x_2\}
\end{aligned} \tag{2.7}$$

then this reveals a sparse dependence between the variables, in this case that the variables

form a *chain*. For the factorization in (2.7), it can be shown that the marginalization can be done in $\mathcal{O}(n|\mathbb{X}|^2)$ computations, a large improvement over the general exponential complexity. This form of structure follows from what is called *conditional independence*. More generally, consider the disjoint set of variables X, Y and Z. We say that X and Y are conditionally independent of each other given Z if

$$\begin{aligned}\mathsf{Pr}\{\mathsf{X},\mathsf{Y},\mathsf{Z}\} &= \mathsf{Pr}\{\mathsf{Z}\}\,\mathsf{Pr}\{\mathsf{X}\mid\mathsf{Z}\}\,\mathsf{Pr}\{\mathsf{Y}\mid\mathsf{Z},\mathsf{X}\} &(2.8)\\ &= \mathsf{Pr}\{\mathsf{Z}\}\,\mathsf{Pr}\{\mathsf{X}\mid\mathsf{Z}\}\,\mathsf{Pr}\{\mathsf{Y}\mid\mathsf{Z}\} &(2.9)\end{aligned}$$

and we write $\mathsf{X}\perp\!\!\!\perp\mathsf{Y}\mid\mathsf{Z}$. In terms of graphs this is also called *graph separation*. In conclusion, our ability to perform efficient marginalization of joint probability distributions hinges on such sparse structures.

2.2 Graphical models

The following section will introduce two graphical representations used for probability distribution in this thesis, the *factor graph* representation and the *Markov Random Field* (MRF) representation.

2.2.1 Factor graphs

A factor graph is a type of *bipartite graph* consisting of *variable nodes* and *factor nodes*, i.e. edges are only between variable and factor nodes, and was first described in [7]. A factor graph represents a function $f(x_1,\ldots,x_n)$ that can be factorized as

$$f(x_1,\ldots,x_n)=\prod_a f_a(x_{\mathsf{N}(a)}) \tag{2.10}$$

where the expression $f_a(x_{\mathsf{N}(a)})$ denotes a *factor* of the function f, $\mathsf{N}(a)$ represents the set of neighbors of the factor node a, and $x_{\mathsf{N}(a)}$ represents the set of neighboring variable nodes of the factor f_a.

When used for inference, factor graphs will in general not model a true probability distribution, but instead a function *proportional* to it, such that

$$f(x_1,\ldots,x_n)\propto p(x_1,\ldots,x_n). \tag{2.11}$$

The *normalization constant* or *partition function*, often denoted by Z when unambiguous,

can then be computed from

$$Z = \sum_{x_1,\ldots,x_n} f(x_1, \ldots, x_n). \tag{2.12}$$

The function f will otherwise satisfy everything else that p must satisfy.

To illustrate a factor graph, consider the distribution $p(x, y, z) = p(z)p(x|z)p(y|z)$. A natural factorization is $f(x, y, z) = f(z)f(x, z)f(y, z)$ which is illustrated in Figure 2.1.

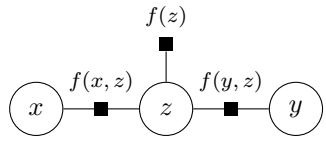


Figure 2.1: *Factor graph illustration of the function* $f(x, y, z) = f(z)f(x, z)f(y, z)$.

2.2.2 Markov random fields

We will here describe an alternative graphical representation called *Markov random fields* (MRF). An MRF follows very much the same structure as a factor graph with the distinction that we in general factorize the function into the maximal *cliques* of the MRF, which can be written as

$$p(x_1, \ldots, x_n) \propto \prod_{\mathcal{C} \in \mathscr{C}} \psi_{\mathcal{C}}(x_{\mathcal{C}}) \tag{2.13}$$

where $\mathcal{C}$ denotes a maximal clique and $\mathscr{C}$ the set of maximal cliques. In terms of graphs, a clique is defined as a set of nodes where all nodes are connected to all other nodes by an edge. We will assume that we can further factorize the function into *node potentials* and *edge potentials*, namely that we can write

$$p(x_1, \ldots, x_n) \propto \prod_{i \in \mathcal{V}} \phi_i(x_i) \prod_{(j,k) \in \mathcal{E}} \psi_{jk}(x_j, x_k) \tag{2.14}$$

where $\mathcal{V}$ denotes the index set of the nodes in the MRF, the *vertex index set*, and $\mathcal{E}$ the *edge index set*. An illustrative example of the MRF equivalent of the factor graph in Figure 2.1 can be found in Figure 2.2. A better illustrative example of how the two representations differ can instead be found in Figure 2.3 where the variables x, y and z form a clique. For a clique size of 3, this will be the case when there is no structure between the variables to exploit.

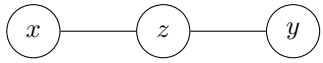


Figure 2.2: *Markov random field illustration of the function* $f(x, y, z) = f(z)f(x, z)f(y, z)$.

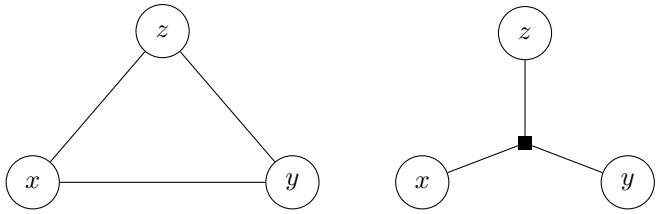


Figure 2.3: *The difference between how an MRF and factor graph represent a clique of size* 3.

2.3 Approximate marginal computation by loopy belief propagation

As discussed in Chapter 2.1.1, we depend on exploitable structure between the latent variables of our system for efficient inference. In particular, if the structure between the latent variables allows us to graphically represent the dependencies as a *tree*, the variables obey a specific *local dependency* between each other. In this case, we are able to use a favorable algorithm called *Belief propagation* (BP). The BP algorithm was first introduced and described by Pearl in [5].

In real-world scenarios, unfortunately, it is common to use probability distributions that have *cycles*, thus violating the tree representation constraint. However, due to the local dependency between the variables assumed by BP, it is still possible to perform BP on such a "loopy" graph. Doing this we arrive at the method known as LBP, which was in fact suggested by Pearl himself [5]. It has been remarkably successful in approximating the exact solution for many problems, such as target tracking [8], [10], [12]–[17], [19], [20]. One of the earliest examples of this success was in the context of error-correcting *turbo codes* [21].

2.3.1 Iterative message passing

We will here elaborate on how the LBP works and how it is based on the local dependency assumption exploited by the BP algorithm. Typically, in the application of LBP on loopy graphs, exact computation of the true marginals of the original joint probability distribution is not performed. Instead, what is computed are *beliefs*. The term "belief" is used to describe these functions to distinguish them from ordinary marginals, as beliefs

are not always the true, consistent marginals derived from a joint probability distribution. However, they behave like marginals in the sense that they share similar properties, such as being non-negative and adding up to 1. In this thesis, however, the two terms "marginal" and "belief" will still be used interchangeably. The beliefs are given by the *messages* between the variables in the graph. These messages are, for discrete factor graphs, given by the iterative application of the equations

$$\mu_{a\to i}(x_i) \leftarrow \sum_{x_{\mathsf{N}(a)\setminus\{i\}}} f_a(x_{\mathsf{N}(a)}) \prod_{j\in\mathsf{N}(a)\setminus\{i\}} \mu_{j\to a}(x_j), \tag{2.15}$$

$$\mu_{i\to a}(x_i) \leftarrow \prod_{b\in\mathsf{N}(i)\setminus\{a\}} \mu_{b\to i}(x_i) \tag{2.16}$$

where $\mathsf{N}(i)$ and $\mathsf{N}(a)$ denotes the set of neighbors to variable i and factor a, respectively, $\sum_{x_{\mathsf{N}(a)\setminus\{i\}}}$ denotes the iterated sum over all values of $x \in \mathsf{N}(a) \setminus \{i\}$, and the message from factor a to variable i is denoted as $\mu_{a\to i}(x_i)$ while the message in the opposite direction is denoted as $\mu_{i\to a}(x_i)$. Note that the equations in (2.15) and (2.16) are only given up to scale, which will be relevant later.

After the messages converge, the beliefs $\hat{p}(x_i)$ can be calculated as

$$\hat{p}(x_i) \propto \prod_{a\in\mathsf{N}(i)} \mu_{a\to i}(x_i) \tag{2.17}$$

where the proportionality sign indicates we need to normalize the beliefs to ensure that they indeed sum to 1 as required.

2.3.2 Loopy belief propagation as a variational inference method

The fact that message passing can be used on loopy graphs and, more importantly, that the estimates we get often are useful, was initially unjustified from a theoretical point of view. In the nominal work by Yedidia et. al [22], [23] they relate the LBP approximation to the constrained optimization of the *Bethe free energy* function F_B that occurs in statistical mechanics. By rewriting our original probability distribution $p(x_1, \ldots, x_n)$ as

$$p(x_1, \ldots, x_n) = \frac{1}{Z} e^{-E(x_1,\ldots,x_n)}, \tag{2.18}$$

$$E(x_1, \ldots, x_n) = -\sum_a \ln f_a(x_a), \tag{2.19}$$

where Z denotes the normalization constant of the factor graph representation of p and a denotes the factors of p, the Bethe free energy function F_B takes the form

$$F_B = U_B - H_B, \tag{2.20}$$

$$U_B = \mathbb{E}_q[E(x_1, \ldots, x_n)] \tag{2.21}$$

$$H_B = -\mathbb{E}_q[\ln q(x_1, \ldots, x_n)] \tag{2.22}$$

where q is the *surrogate* function for p, $\mathbb{E}_q[\bullet]$ denotes the expectation under the density q, U_B denotes the *variational average energy* and H_B the *variational entropy*.

The surrogate function q is supposed to approximate the original probability distribution $p(x_1, \ldots, x_n)$ as a function that also obeys a factorization which allows for efficient marginalization. The *Bethe approximation* [24] that the Bethe free energy function uses is that the surrogate function q can be factorized according to

$$q(x_1, \ldots, x_n) = \frac{\prod_a q_a(x_{\mathsf{N}(a)})}{\prod_{i=1}^n q_i(x_i)^{d_i - 1}} \tag{2.23}$$

where the symbol $\prod_a$ indicates the product over subsets of variables that constitute the factors a of our original distribution p, and d_i represents the degree of node i, which refers to the number of edges adjacent to it. This particular choice of factorization is exact for probability distributions that allow for a tree representation. Thus, the surrogate function q that solves the optimization can be interpreted as the closest "tree-like distribution" of our original distribution. Note that, since the factorization in (2.23) is over the same nodes and edges as in our original distribution, the surrogate function itself will not be a proper probability distribution in general. We can show, however, that in the case that $q(x_1, \ldots, x_n)$ indeed is a proper joint distribution, then [23]

$$F_B = D(q||p) + F_H \tag{2.24}$$

$$\implies F_B \geq F_H, \tag{2.25}$$

where $D(q||p)$ is the *Kullback-Leibler divergence* between the distributions q and p. The inequality in (2.25) follows from the fact that $D(q||p) \geq 0$ and $D(q||p) = 0$ iff $q = p$. The functions $q_a(x_{\mathsf{N}(a)})$ and $q_i(x_i)$ that appear in (2.23) will, on the other hand, in the same way as the beliefs from LBP *behave* like probability distributions after introducing the necessary constraints in the optimization to ensure consistent behavior. Namely, the

constraints that are used is that each variable belief q_i must sum to 1,

$$g_i(x_i) = \sum_{x_i} q_i(x_i) - 1, \tag{2.26}$$

that each factor belief q_a must sum to 1,

$$g_a(x_a) = \sum_{x_a} q_a(x_a) - 1, \tag{2.27}$$

and that variable beliefs are retrievable by marginalization of the factor beliefs

$$g_{a\to i}(x_i) = \sum_{x_a \setminus x_i} q_a(x_a) - q_i(x_i) \tag{2.28}$$

where the notation $x_a \setminus x_i$ is used to indicate all variables of factor a except x_i. Strictly speaking, the constraint that each belief has to be non-negative is also necessary. They argue in [23] that this constraint is inactive at the fixed-point of LBP such that it makes no difference to the optimization and is therefore neglected. The constrained optimization problem can be solved by using the *Lagrangian* $\mathscr{L}$ [25] which augments the objective function F_B with the constraints in (2.26) to (2.28) and the corresponding *Lagrangian multipliers* λ_i, λ_a and $\lambda_{a\to i}$. The result is that the Lagrangian can be written as

$$\begin{aligned}\mathscr{L} = F_B &- \sum_a \lambda_a g_a(x_a) - \sum_i \lambda_i g_i(x_i) \\ &- \sum_{i'} \sum_{a\in \mathsf{N}(i')} \sum_{x_{i'}} \lambda_{a\to i'} g_{a\to i'}(x_{i'}).\end{aligned} \tag{2.29}$$

where i' denotes the indices over all variable nodes with degree $d_i \geq 2$, the constraints g_a and g_i with corresponding multipliers λ_a and λ_i are called the *node constraints* and $\lambda_{a\to i}$ and g_{a_i} the *edge constraints*. The node constraint multipliers are scalars that are normalized away in the end, and are of little interest. More importantly, the insight they show in [23] is that the edge constraint multipliers $\lambda_{a\to i}$ on the other hand are related to the messages $\mu_{a\to i}$ as

$$\lambda_{a\to i}(x_i) = \ln \prod_{b\in \mathsf{N}(i)\setminus\{a\}} \mu_{b\to i}(x_i) \tag{2.30}$$

when evaluating the Lagrangian in (2.29) at a stationary point of the function, proving that the LBP beliefs are the optimal solution to the Bethe approximation.

2.3.3 The Bethe constant and the Bethe pseudodual

Another useful aspect of the Bethe approximation is to estimate the normalization constant of the original probability distribution. This follows from the fact that the Bethe free energy function is based on the *Helmholtz free energy* F_H which is defined as

$$F_H = -\ln Z. \tag{2.31}$$

Analogous to (2.31), the *Bethe constant* Z_B is then related to the Bethe free energy as

$$F_B = -\ln Z_B \tag{2.32}$$

$$\implies Z_B = \exp(-F_B). \tag{2.33}$$

This implies that we can use the fixed-point messages from LBP to form q and insert it appropriately into (2.33) to estimate the normalization constant Z. We note that this also implies that, if the Bethe approximation is a true distribution such that the inequality in (2.25) is satisfied, then

$$Z_B \leq Z, \tag{2.34}$$

i.e., the Bethe constant is *guaranteed* to underestimate the true normalization constant. A more exotic quantity to compute is the *Bethe pseudodual* function [26], [27]. This function is the Lagrangian of the constrained optimization problem that LBP solves where the optimal q is inserted and instead the Lagrangian multipliers are variables. The fact that we are able to know this optimal q before-hand stems from the simple fact that we know they are given by the LBP equations, specifically (2.17), which are exclusively given by the LBP messages. Thus, when inserting the LBP messages for the LBP beliefs together with the Lagrangian multipliers, we get a function entirely in terms of the factors of our factor graph and the LBP messages. In the sequel we will use a slightly modified Lagrangian where we only keep the edge constraints, as is done in [26]. We can justify doing this by inserting normalized beliefs in the pseudodual, where we instead let the normalization constants be functions of the LBP messages. We then get that the node constraints $g_a(x_a)$ are $g_i(x_i)$ in (2.26) and (2.27) vanish for all iterations of LBP.

2.3.4 Properties of loopy belief propagation

The following section intends to summarize selected research results on the properties of LBP, and in particular regarding data association in MTT. Firstly, there are no guarantees for the accuracy of the approximate marginals, or that the algorithm even converges, a

fact that had been dismissed previously. In the case where the LBP algorithm does not converge it instead *oscillates*, i.e., the messages start oscillating.

Understanding LBP behavior theoretically is in the general case hard, and so most of the results are either empirical or theoretical, but on specific graphs where assumptions about the structure is made. In an empirical study by Murphy et. al [28], which was about the first of its kind, they try to understand the behavior of LBP in a more general setting outside of the already established error-correcting code context. Their findings are that LBP seems to compute accurate marginals for the cases where it converges. When LBP oscillates they are able to achieve convergence by introducing *momentum* in the message update, meaning the messages are instead updated as a weighted average of the full update and the previous message. Despite achieving convergence after this modification, the beliefs remains inaccurate in general. What they do observe is that small priors and weights for parameters in the graph affect the convergence. They speculate that this can be related to how likely the observations are, but further testing seems to disagree with this. Lastly, they state that in their results it seems that whether LBP estimates accurate marginals or not is not related to the size of the loops in the graph.

In [29] by Ihler et. al, they do a theoretical study of convergence conditions based on two different measures of error accumulation under LBP. In particular, they derive sufficient convergence conditions for LBP convergence from the the largest ratio of the factors in the graph, called the *dynamic range*, as long as it is finite.

Two particularly important references for the work in this thesis are those of Vontobel [26] and Williams et. al [8]. In both references they inspect the use of LBP on a specific bipartite graph. Although the underlying problem they try to solve are related, the motivation for doing this in [26] is to estimate the permanent of certain matrices, while in [8] it is to estimate association marginals in a single-cluster, single-hypothesis multi-target tracking scenario. For these kind of graphs they both show guaranteed convergence of LBP to a unique fixed-point, regardless of message initialization. Relating this to the Bethe free energy function, [26] attributes this to the fact the Bethe free energy function of the graph is indeed *convex*. In order to measure convergence they use in [8] a dynamic range measure similar to the one defined in [29] as a distance metric between messages in two different iterations, and terminate when this gets below some threshold. In [26] they use the Bethe pseudodual function described previously in Chapter 2.3.3 evaluated in two different iterations as the distance metric between messages. In the results of [8] they observe that for tracking cases where the misdetection probability is low, i.e. that there is a large *Signal-to-noise ratio* (SNR), LBP tends to struggle. This could be related to similar observations in [28]. Another important result from [26] is the fact

that they show that *the Bethe approximation for this particular graph indeed is a proper joint distribution*, and so we are guaranteed that the associated Bethe constant always underestimates the true normalization constant.

Lastly, in one of the latest works by Williams et. al [30] they present an approximate marginalization method for association graphs similar to the one they present in [8]. The difference is that the graph is generalized for *multi-scan* data association, i.e. where they have multiple sets of measurements, either from multiple sensors or over multiple timesteps. Due to the Bethe free energy function of the multi-scan association graph being nonconvex, they experience that running LBP directly on this graph could have undesirable effects in terms of e.g. convergence. Therefore, they instead modify the Bethe free energy function by using the *fractional free energy* where they scale a certain part of the free energy function to make it convex. They then derive an LBP-like algorithm for this approximation to achieve better and more robust performance.

3 | Concepts in multi-target tracking

In order to do multi-target tracking, as with all forms of estimation, a foundation of concepts and mathematical models must be established. The following chapter intends to provide an overview of the standard, general concepts and models that are common in modern tracking literature [31]–[34] and properly define them. Before proceeding we orient the reader that a major part of the material in the following chapter will be either similar or equal to parts of the overview provided in the preceding project report.

3.1 Multiple measurements and hypothesis enumeration

Perhaps the most fundamental concept in MTT is the notion of receiving multiple measurements that can originate from *different targets*. The very reason we concern ourselves with association hypotheses, which we so far only have introduced conceptually, is precisely because we have to consider all possible origins for the received measurements, at least if we seek the optimal solution. The procedure of finding all possible origins for the received measurements is called *hypothesis enumeration*, and is a serious bottleneck in any MTT filter. In fact, this very reason is why we in general have two types of MTT filters – *single-hypothesis* and *multi-hypothesis*. In single-hypothesis MTT filters we approximate the estimated track states by marginalizing over the enumerated hypotheses in a timestep. A classical example of such a filter is the celebrated *Joint probabilistic data association* (JPDA) filter, first published by Fortmann et. al in [31]. This of course trades accuracy for computational performance since a considerable amount of information is lost in the marginalization procedure. As previously mentioned, in this work we instead consider multi-hypothesis tracking, where we keep a collection of hypotheses

across timesteps. Our motivation for doing this is to be able retain more information to dynamically adapt to different scenarios for more reliable and robust tracking.

3.1.1 Extended object tracking and the at-most-one assumption

While on the topic of hypothesis enumeration and measurement origin, a discussion about the *at-most-one assumption* that is made is warranted. This assumption states 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 [35].

3.2 Definitions

In multi-hypothesis, multi-target tracking literature there are two concepts in particular that vary in how they are defined, which is the notion of a *track*, and how it differs from a target, and a *hypothesis*. The following section will give the formal, mathematical definition that is used in this thesis. Additionally, we will define what is meant by a *cluster*.

3.2.1 Track

We will define a *track* as a sequence of measurements, i.e. detections, 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

$$\mathcal{I}^t = [i_1, \dots, i_k] \tag{3.1}$$

where $i_l \in \{0, \dots, m_l, N\}$ for each $l \in \{1, \dots, k\}$, where index i_l corresponds to measurement $\boldsymbol{z}_l^{i_l}$ for $0 < i \leq m_l$, misdetection when $i = 0$ and *nonexistence* when $i_l = N$ to indicate a track that has not been detected yet, and as such "does not exist".

To exemplify this, assume we have over three timesteps received the measurement sets

$$Z_1 = \{\boldsymbol{z}_1^1\},$$
$$Z_2 = \{\boldsymbol{z}_2^1\},$$
$$Z_3 = \{\boldsymbol{z}_3^1\}.$$

In timestep 3, one possible set of tracks can then be

$$\mathcal{I}^1 = [1, 1, 0]\,,$$
$$\mathcal{I}^2 = [N, N, 1]\,,$$

which indicates the following. Based on the three measurements, we hypothesize that we have two tracks. Track 1 was detected in the first two timesteps and misdetected in the third. We were not aware that track 2 existed until timestep 3, but initialized it then as the measurement was not associated to track 1.

Target and track distinction

We distinguish between a target and track by saying that a target remains semantically exactly one object in the real world, while we can have multiple tracks for the same target. Therefore, one can think of a track as a possible trajectory of a target given the measurements we have, and that there are multiple ways of interpreting the measurements we have, hence multiple tracks.

3.2.2 Association hypothesis

The full definition we use for an association hypothesis can be found in [1]. To summarize, we notate a hypothesis in timestep k as $\theta_{1:k}$, where the subscript $1:k$ indicates that the hypothesis is based upon a sequence of measurement sets from timestep 1 through k. It will be useful to indicate that a track t exists conditioned on the hypothesis $\theta_{1:k}$, and so we will allow, based on the definition of a track in Chapter 3.2.1, the alternative definition that a hypothesis can be represented as a subset of all the n_k track indices that exist in timestep k,

$$\theta_{1:k}^r \subseteq \{1, \ldots, n_k\}, \tag{3.2}$$

such that we write $t \in \theta_{1:k}$ and say that track t exists in hypothesis $\theta_{1:k}$ and conversely for $t \in \theta_{1:k}$. If we consider the example in Chapter 3.2.1, the association hypothesis is

$$\theta_{1:3}^{1} = \{1, 2\}$$

where we arbitrarily chose to enumerate the hypothesis as hypothesis 1. Note that we will we will overload the superscript to either denote which hypothesis it is out of the possible hypotheses or denote which cluster the hypothesis variable belongs to. Thus, the symbol $\theta_{1:k}^{l}$ can be used to indicate the l^{th} hypothesis in a cluster of tracks, while $\theta_{1:k}^{c}$ indicates the set of hypotheses in cluster c. It will be clear from the context what is meant and if not, explicitly stated.

3.3 Track dynamics model

We will keep the assumptions necessary to do track state estimation as general as possible until otherwise required. We notate the state vector of a particular track by $\boldsymbol{x}_k^t$ to indicate the state of track t in timestep k and use $\boldsymbol{x}_k$ when it is not necessary to indicate a particular track. The prior distribution for track t is denoted by $p_k^t(\boldsymbol{x}_k^t|Z_{1:k-1})$ and we assume the state dynamics obey a *Markov model* given by the transition model $p_x(\boldsymbol{x}_k^t|\boldsymbol{x}_{k-1}^t)$. Lastly, we assume that targets *depart* from the surveillance region with some probability given by the probability distribution $P_S(\boldsymbol{x})$.

3.4 Track measurement model

We assume the measurement model has the form $p_z(\boldsymbol{z}_k|\boldsymbol{x}_k^t)$ for $\boldsymbol{z}_k \in Z_k$. From this the *likelihood* is given as

$$p(\boldsymbol{z}_k|Z_{1:k-1}) = \int p(\boldsymbol{x}_k|Z_{1:k-1}) p_z(\boldsymbol{z}_k|\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{x}_k. \tag{3.3}$$

In particular, we use the notation

$$l^{jt} = \int p_k^t(\boldsymbol{x}_k^t|Z_{1:k-1}) p_z(\boldsymbol{z}_k^j|\boldsymbol{x}_k^t)\,\mathrm{d}\boldsymbol{x}_k^t \tag{3.4}$$

to indicate the value of the likelihood when assuming a particular track and evaluating it in a measurement $\boldsymbol{z}_k^j$ with $j \in \{1, \ldots, m_k\}$ for m_k measurements in timestep k. Lastly, we assume that each target is detected with a probability given by the probability distribution $P_D(\boldsymbol{x}_k)$ and therefore *misdetected* with probability $1 - P_D(\boldsymbol{x}_k)$.

3.4.1 Unassigned measurements

When considering the origin of a measurement, we have to consider the possibility that *a measurement does not originate from any existing track*. We call these measurements *unassigned*, and we consider here the two standard origins, *clutter* and *undetected targets*.

Clutter

Clutter in a measurement scan is also called *false alarms*, and as the name suggests are measurements that shall be discarded as noise. In practice this can originate from e.g. a sensor fault or a briefly passing object that we should not initialize a track for. Mathematically, we assume that the number of clutter measurements φ_k in measurement scan Z_k can be modelled as a *Poisson distribution*, such that

$$\mathsf{Pr}\{\varphi_k\} = e^{-\Lambda}\frac{\Lambda^{\varphi_k}}{\varphi_k!} \tag{3.5}$$

where

$$\Lambda = \int \lambda(\boldsymbol{z}_k)\,\mathrm{d}\boldsymbol{z}_k, \tag{3.6}$$

and $\lambda(\boldsymbol{z})$ is the *intensity* of the distribution. Since clutter is by definition random noise, we assume it is independent of other undetected targets and target detections.

Undetected targets

As a natural consequence of doing multi-target tracking, we have to allow the possibility that *new targets enter our surveillance region*. Because of this, there will in each timestep be an unknown number of undetected targets that we can detect. Modelling the distribution over undetected targets will be deferred to Chapter 4, but to facilitate this we will here model the distribution for number of new targets β_k that arrive in the surveillance region. Namely, we assume β_k to be Poisson distributed with distribution

$$\mathsf{Pr}\{\beta_k\} = e^{-\mathrm{M}}\frac{\mathrm{M}^{\beta_k}}{\beta_k!} \tag{3.7}$$

where

$$\mathrm{M} = \int \mu(\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{x} \tag{3.8}$$

and $\mu(\boldsymbol{x}_k)$ is the intensity of the distribution over the state space of tracks.

3.5 Gating of measurements

In practice we will not consider all measurements for each track t, but instead all *gated* measurements $\boldsymbol{z}_k^j$, for computational reasons. Theoretically speaking, this is a matter of computing the associated likelihood l^{jt} and setting it to 0 if it is below some threshold g. In terms of hypothesis enumeration, this removes all hypotheses from the hypothesis space where the track is associated to the measurement, which significantly reduces its size. The region of valid measurements, be it the entire measurement space or the part that passes the gating, is called the *validation gate*. We will see that the probability that a track t should associated to measurement j becomes negligible for small l^{jt}, and so this is a justified design choice when comparing the accuracy loss to the computational improvement. An example of gating can be found in Figure 3.1, which identical to the illustration in Figure 3.1 in the preceding project report.

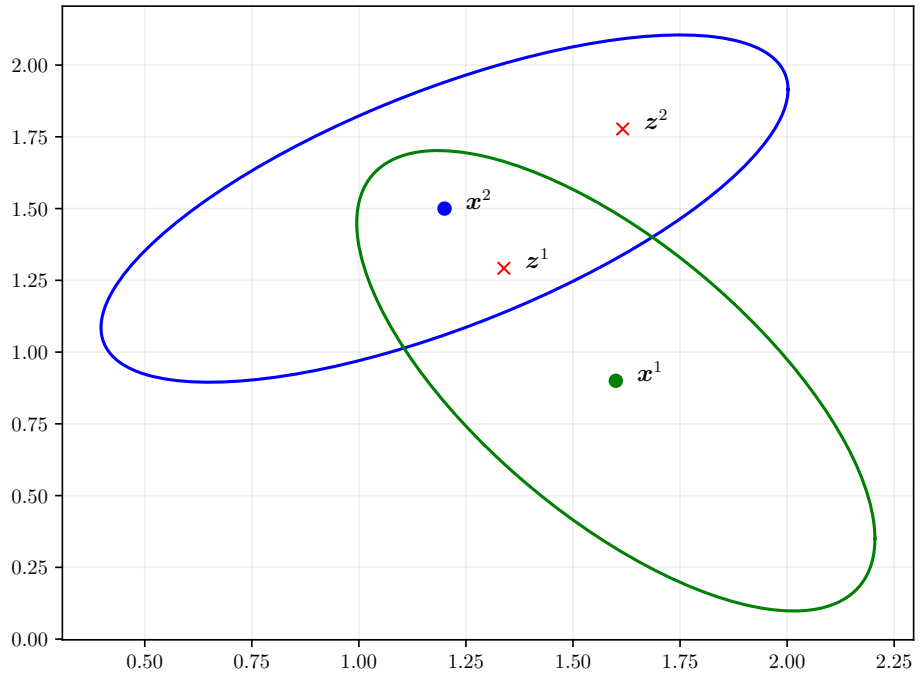


Figure 3.1: *An example with two tracks and two measurements, denoted by $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ and $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$, respectively. The validation gate for each track is visualized as an ellipsis, such that all measurements that are outside the ellipsis are discarded as potential detections of the corresponding track. This means e.g. that measurement $\boldsymbol{z}^2$ is not gated by track $\boldsymbol{x}^1$ and that both $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ are gated by $\boldsymbol{x}^2$.*

3.6 Cluster

A *cluster* in MTT is defined as a collection of tracks that are linked together by measurements that lie in the intersection of their validation gates [31]. When multiple tracks gate the same measurement, the data association problem has to be solved jointly as a single cluster as the track associations become dependent on each other. Conversely, this implies that two different clusters are by definition *independent* of each other. The independence property means we can do data association for each cluster separately, which is significantly more efficient than to solve the entire association problem jointly for all tracks. We refer to data association problems that are strictly internal to one cluster as *single cluster* while when we do it for all the clusters as *multi-cluster*. In a multi-hypothesis setting, each cluster maintains its own hypothesis distribution where each track in the cluster must exist in at least one of these hypotheses. A visualization of two clusters can be found in Figure 3.2.

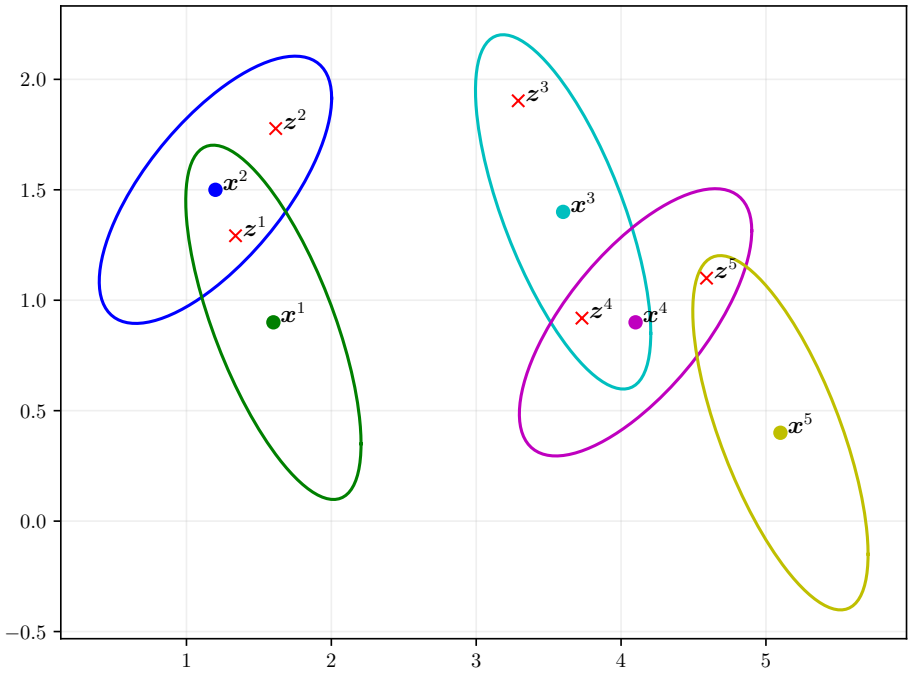


Figure 3.2: *Association case with two clusters. The left-most cluster is generated because tracks $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ both gate measurement $\boldsymbol{z}^1$. The right-most cluster is generated because tracks $\boldsymbol{x}^3$ and $\boldsymbol{x}^4$ both gate measurement $\boldsymbol{z}^4$ and tracks $\boldsymbol{x}^4$ and $\boldsymbol{x}^5$ both gate measurement $\boldsymbol{z}^5$. Notice that for the right-most cluster, tracks $\boldsymbol{x}^3$ and $\boldsymbol{x}^5$ do not share any measurement, but are still dependent on each other through track $\boldsymbol{x}^4$.*

3.6.1 Cluster merging

An important part of any MTT pipeline is *cluster management*, and in particular *cluster merging* which was mentioned in the introduction. We reiterate here this challenge in slightly more technical detail.

Cluster merging happens when multiple clusters *interact* due to tracks in separate clusters gating the same measurements. In multiple hypothesis tracking, a brute-force approach to merge the clusters into one is to take the Cartesian product of the collections of prior hypotheses. After cluster merging, the prior hypotheses will then be the members of this Cartesian space, where the probability of each element is the product of the probabilities of the combined elements.

In practice, this approach has a fatal flaw. Considering in a practical implementation one typically keeps the 100 – $150 \sim 10^2$ most likely hypotheses from each timestep, the size of the new prior hypothesis posterior space is in the magnitude $\sim 10^{2M}$ elements for M interacting clusters. Such a large prior hypothesis posterior space makes the posterior hypothesis posterior cluster space in general astronomically large. Hence, keeping the 100 – 150 best hypotheses prunes almost all of the event space, potentially resulting in a large loss of accuracy and more catastrophically, pruning the correct hypothesis.

4 | The Poisson Multi-Bernoulli Mixture filter

The most common and famous multi-target filters today are arguably JPDA, published by Fortmann et. al in [31], and *Multiple hypothesis tracker* (MHT), published by Reid in [36]. In particular, MHT was the first contribution towards an optimal solution for a *multi-hypothesis*, multi-target tracking filter, contrary to JPDA which for each timestep combines the posterior hypotheses into a single hypothesis.

Another breakthrough in multi-target tracking came with the PHD filter by Mahler [9] which built upon FISST that extends probability theory to *sets* of random vectors, where also the cardinality of the set is a stochastic variable. A full introduction is outside the scope of this thesis, and the reader is instead referred to references like [9], [33]. Formulated in the FISST framework, Williams derived in [32] what has later become known as the PMBM filter, which aims to provide a general, optimal solution to the standard multi-target tracking scenario under certain, standard model assumptions. The following chapter intends to explain the building blocks of the PMBM filter. Doing this will yield the necessary expressions for later forming the *joint association posterior* that we require to perform inference. Additionally, this introduction is a more natural stepping stone to motivate one of the main uses for efficient marginalization in a practical application, *track recycling*, which will be discussed in more detail in Chapter 4.6. The intent of this chapter is therefore not to give a thorough deep-dive into the theory that PMBM builds upon, but rather state the necessary results.

The following will present the PMBM in its *Probability generating functional* (PGFL) form, which builds upon the theory of FISST. The main purpose of formulating the multi-object Bayes filter in terms of PGFLs is that the PGFL representation of a given set density is in general more compact and tangible to work with than the set density itself, and so instead we can derive the necessary prediction and update equations we require

by formulating them with PGFLs and then do *functional derivatives*.

4.1 Probability generating functionals

In traditional probability theory, a *Probability generating function* (PGF) is a transformation of a discrete probability distribution over nonnegative numbers to a function that return the probability of events by differentiation. For a discrete probability distribution $\mathsf{Pr}\{x\}$ for the stochastic variable X, its corresponding PGF $G_X(h)$ is given by

$$G_X(h) = \sum_{x=0}^{\infty} h^x \mathsf{Pr}\{x\}\,. \tag{4.1}$$

The probability $\mathsf{Pr}\{X = k\}$ of event $x = k$, $k \in \{0, 1, \dots\}$ is then

$$\mathsf{Pr}\{k\} = \frac{\mathrm{d}^k}{\mathrm{d}h^k} G_X(h)\bigg|_{h=0}. \tag{4.2}$$

From (4.1) we can show that for two independent stochastic variables X and Y with PGFs $G_X(h)$ and $G_Y(h)$, respectively, the PGF of their sum $Z = X + Y$ is simply

$$G_Z(h) = G_X(h)G_Y(h). \tag{4.3}$$

The probability distribution $\mathsf{Pr}\{Z\}$ becomes a convolution over $\mathsf{Pr}\{X\}$ and $\mathsf{Pr}\{Y\}$, which often times is more complicated to compute and use than the simple product in (4.3), showing the main strength of the representation

Before generalizing this notion to set densities and set integrals, let us briefly introduce what is meant by a *Random finite set* (RFS) for the purpose of this text. Let X be an RFS with $\mathbb{R}^d$ as its *base space*. This means that a *realization* X of X can be $X = \{\boldsymbol{x}^1, \dots, \boldsymbol{x}^n\}$ with $\boldsymbol{x}^i \in \mathbb{R}^d$, $i \in \{1, \dots, n\}$ and each $\boldsymbol{x}^i$ distributed according to some distribution $p(\boldsymbol{x})$. Additionally, the cardinality n is also a stochastic variable with a distribution that can be found from the set density $f(\mathsf{X})$ to X. With this, we define the PGFL $G_{\mathsf{X}}[h]$ for the RFS as

$$G_{\mathsf{X}}[h] = \int h^X f(X)\delta X \tag{4.4}$$

where we use the bracket notation $[h]$ in $G_{\mathsf{X}}[h]$ to emphasize that $G_{\mathsf{X}}[h]$ is a *functional* that operates on some *test function* $h(\boldsymbol{x}) : \mathbb{R}^d \to [0, \infty)$, the notation h^X is defined as

$$h^X = \prod_{\boldsymbol{x} \in X} h(\boldsymbol{x}) \tag{4.5}$$

with $h^{\emptyset} = 1$, $f(X)$ is the set density for X evaluted in the set X and $\int f(X)\delta X$ is the set integral of $f(X)$ from FISST theory where we use X instead of X to indicate that we evalute the function in all possible realizations X when doing the integral. For independent RFSs X and Y, the PGFL of their union $\mathsf{Z} = \mathsf{X} \cup \mathsf{Y}$ is given as

$$G_{\mathsf{Z}}[h] = G_{\mathsf{X}}[h]G_{\mathsf{Y}}[h], \tag{4.6}$$

analogous to (4.3).

Lastly, we will state the *functional derivative* of the PGFL $G[h]$ with respect to $X = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n\}$ as

$$\frac{\delta G}{\delta X} = \frac{\delta^n G}{\delta \boldsymbol{x}^1 \ldots \delta \boldsymbol{x}^n} \tag{4.7}$$

where each derivative with respect to an element of X is defined as

$$\frac{\delta G}{\delta \boldsymbol{x}} = \lim_{\varepsilon \to 0^+} \frac{G[h + \varepsilon \delta_{\boldsymbol{x}}] - G[h]}{\varepsilon} \tag{4.8}$$

where $\delta_{\boldsymbol{x}}$ is the *Dirac-delta function* centered at $\boldsymbol{x}$, and we define $\frac{\delta G}{\delta \emptyset} = G[h]$. Notice also that we indicate the realization X in (4.7) as the recursion depends on its cardinality.

From (4.7) we can derive the *fundamental theorem of multi-object calculus*

$$f(X) = \frac{\delta G}{\delta X}[0] \tag{4.9}$$

where a proof can be found in [9]. The result in (4.9) is what allows us to derive tangible expressions for achieving multi-target filtering based on FISST, and will play a central role in the sequel.

4.2 Constructing the prior

In the following, we will drop indications of what timestep we are in for notational simplicity. Constructing the PGFL prior in PMBM is based on two arguments. The first argument is that we can model the tracks as *Bernoulli sets*, which obeys the distribution

$$f(X^t) = \begin{cases} 1 - r^t, & X^t = \emptyset, \\ r^t p^t(\boldsymbol{x}^t), & X^t = \{\boldsymbol{x}^t\}, \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

with PGFL

$$G^{\mathrm{b}}[h] = 1 - r^t + r^t p^t[h] \tag{4.11}$$

where $p^t[h]$, more generally $f[h]$ for some function f and test function h, is the *linear functional*

$$f[h] = \int f(\boldsymbol{x}) h(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{4.12}$$

r^t is the existence probability of track t and $p^t(\boldsymbol{x}^t)$ is the track state distribution. We can form a *Multi-Bernoulli* (MB)-component by taking the union over all such tracks that are tracked, which makes the PGFL of the resulting distribution simply the product over their PGFLs

$$G^{\mathrm{mb}}[h] = \prod_{t=1}^{n} G^{\mathrm{b}}[h] \tag{4.13}$$

for n independent tracks. If we now generalize the PGFL in (4.13) for *multiple hypotheses* $\theta^l, l = 1, \ldots, L$ for L hypotheses defined as in Chapter 3.2.2, the resulting PGFL becomes a mixture, or a *Multi-Bernoulli Mixture* (MBM), and has the form

$$G^{\mathrm{mbm}}[h] = \sum_{l=1}^{L} w^l G^{\mathrm{mb}}[h] \tag{4.14}$$

where we introduced the *hypothesis weight* w^l with the only requirement that it is related to the hypothesis distribution $\mathsf{Pr}\{\theta^l\}$ by

$$\mathsf{Pr}\{\theta^l\} \propto w^l \tag{4.15}$$

and omit the hypothesis index l in $G^{\mathrm{mb}}[h]$ for notational simplicity as it should be clear from context, and rather explicitly state it later if ambiguous. In PMBM, however, we write (4.14) in a slightly different form. Consider some particular hypothesis θ^l. We substitute the hypothesis weight w^l with a product over weights for each track existing in the hypothesis

$$w^l = \prod_{t=1}^{n^l} w^{lt} \tag{4.16}$$

for n^l tracks in θ^l, where the weights w^{lt} are given by what associations we make for a given track. The resulting MBM PGFL we use is therefore

$$G^{\mathrm{mbm}}[h] = \sum_{l=1}^{L} \prod_{t \in \theta^l} w^{lt} \left(1 - r^{lt} + r^{lt} p^{lt}[h]\right) \tag{4.17}$$

where r^{lt} indicates the existence probability of track t under the association made in hypothesis l and similarly for the linear functional of the track state distribution $p^{lt}[h]$.

The PGFL in (4.17) only accounts for *detected targets* that we actively track. In Chapter 3.4.1 we mentioned that we deferred modelling *undetected targets*, which we model now. We assume they obey a *Poisson point process* with set density

$$f(X) = e^{-\nu[1]} \prod_{\boldsymbol{x} \in X} \nu(\boldsymbol{x}), \tag{4.18}$$

where $\nu(\boldsymbol{x})$ is the intensity of the process and

$$\nu[1] = \int \nu(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{4.19}$$

The corresponding PGFL is

$$G^{\mathrm{p}}[h] = \exp(\nu[h-1]). \tag{4.20}$$

In conclusion, the PGFL prior that PMBM does filtering on, for all timesteps k, is

$$G_k^{\mathrm{pmbm}}[h] = G_k^{\mathrm{p}}[h] G_k^{\mathrm{mbm}}[h] \tag{4.21}$$

where we have assumed independence between the detected and undetected targets to write the prior as a product. Although we have argued that the PGFL in (4.21) is sound and consistent with our assumptions in Chapter 3, it remains to determine whether it is *useful*. Namely, we desire that the PGFL form is closed under both the prediction and update step, a property called *conjugacy*. Rest assured, as we will see, this is indeed the case for the PGFL in (4.21), further justifying it. The interested reader is referred to [32], [37] for a more elaborate introduction and justification to the above PGFL.

4.3 The prediction step

The prediction step in the PGFL formulation of the Bayes filter is given by [9], [33]

$$G_{k|k-1}[h] = \int G_{X_k|X_{k-1}}[h|X_{k-1}] f_{k-1}(X_{k-1}) \delta X_{k-1} \tag{4.22}$$

where $G_{X_k|X_{k-1}}[h|X_{k-1}]$ is the PGFL of the transition distribution $f_X(X_k|X_{k-1})$ and $f_{k-1}(x_{k-1})$ is the posterior distribution of X_{k-1}. By the assumptions in Chapter 3.3, we

get that the transition distribution for a Bernoulli target X^t is also Bernoulli with PGFL

$$G[h|X_{k-1}^t] = 1 - P_S(\boldsymbol{x}_{k-1}) + P_S(\boldsymbol{x}_{k-1})p_x[h|\boldsymbol{x}_{k-1}] \tag{4.23}$$

where $P_S(\boldsymbol{x}_{k-1})$ is the departure probability for a track with state $\boldsymbol{x}_{k-1}$ and $p_x[h|\boldsymbol{x}_{k-1}]$ is the linear functional of the transition model with respect to $\boldsymbol{x}_k$,

$$p_x[h|\boldsymbol{x}_{k-1}] = \int p_x(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})h(\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{x}_k. \tag{4.24}$$

For independent tracks, we then get that

$$G_{X_k|X_{k-1}}[h|X_{k-1}] = \prod_{\boldsymbol{x}_{k-1}^t \in X_{k-1}} G[h|X_{k-1}^t] \tag{4.25}$$

$$= \prod_{\boldsymbol{x}_{k-1}^t \in X_{k-1}} (1 - P_S(\boldsymbol{x}_{k-1}^t) + P_S(\boldsymbol{x}_{k-1}^t)p_x[h|\boldsymbol{x}_{k-1}^t]). \tag{4.26}$$

such that the prior PGFL over surviving tracks in timestep k is given by

$$G_{k|k-1}^{\text{surviving}}[h] = G_{k-1}[G_{X_k|X_{k-1}}[h|X_{k-1}^t]] \tag{4.27}$$

which follows from inserting (4.25) into (4.22) and using the definition in (4.4).

In order to obey the assumption that we have new, undetected targets arriving in the surveillance region according to a Poisson point process, we additionally need the PGFL $G^{\text{new}}[h]$ which with intensity $\mu(\boldsymbol{x}_k)$ from Chapter 3.4.1 is

$$G^{\text{new}}[h] = \exp(\mu[h-1]). \tag{4.28}$$

Assuming new targets arriving are independent from the surviving tracks, the final prior PGFL is becomes the product of (4.28) and (4.27), yielding

$$G_{k|k-1}[h] = G_{k|k-1}^{\text{surviving}}[h]G^{\text{new}}[h]. \tag{4.29}$$

The final step is to verify that (4.29) can be written on the form in (4.21), which we will here take for granted. Verifying it involves moving the Poisson PGFL $G^{\text{new}}[h]$ into the Poisson component for surviving, undetected targets in $G_{k|k-1}^{\text{surviving}}[h]$ by the *principle of superposition*. This is a property we will use later when we discuss *track recycling* in

Chapter 4.6. We can use (4.7) to show that (4.29) reduces to the tangible functions

$$\nu_{k|k-1}(\boldsymbol{x}_k) = \mu(\boldsymbol{x}_k) + \int p_x(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) P_S(\boldsymbol{x}_{k-1}) \nu_{k-1}(\boldsymbol{x}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1} \tag{4.30}$$

$$w^{lt}_{k|k-1} = w^{lt}_{k-1} \tag{4.31}$$

$$r^{lt}_{k|k-1} = r^{lt}_{k-1} p^{lt}_{k-1}[P_S(\boldsymbol{x}_{k-1})] \tag{4.32}$$

$$p^{lt}_{k|k-1}(\boldsymbol{x}_k) = \frac{\int p_x(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) P_S(\boldsymbol{x}_{k-1}) p^{lt}_{k-1}(\boldsymbol{x}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1}}{p^{lt}_{k-1}[P_S(\boldsymbol{x}_{k-1})]} \tag{4.33}$$

where $p^{lt}_{k-1}(\boldsymbol{x}_{k-1})$ is the posterior state distribution for track t in hypothesis l from the previous timestep $k-1$.

4.4 The update step

The posterior is constructed by first defining the joint PGFL

$$F[g, h] = \int h^{X_k} G_k[g|X_k] f_{k|k-1}(X_k) \delta X_k \tag{4.34}$$

where

$$G_k[g|X_k] = \int g^{Z_k} f_Z(Z_k|X_k) \delta Z_k \tag{4.35}$$

is the PGFL of the set measurement model $f_Z(Z_k|X_k)$ and Z_k denotes the specific measurement set in timestep k. First, notice that the PGFL in (4.34) can interpreted as the PGFL of the joint set distribution $f(\mathsf{X}, \mathsf{Z})$. It should therefore be reasonably clear that we retrieve the set posterior $f(X_k|Z_k)$ in timestep k as proportional to $\delta F[0, h]/\delta Z_k$, i.e. the joint posterior evaluated in Z_k, and where the proportionality constant is given from the set likelihood $\int f_Z(\mathsf{Z}|X_k) f(X_k) \delta X_k = f(\mathsf{Z})$ which can be shown to have PGFL

$$G_k[g] = \int g^{Z_k} f(Z_k) \delta Z_k \tag{4.36}$$

$$= \frac{\delta F[0, 1]}{\delta Z_k}. \tag{4.37}$$

Hence, the update in PGFL form is therefore given by

$$G_k[h] = \frac{\dfrac{\delta F[0, h]}{\delta Z_k}}{\dfrac{\delta F[0, 1]}{\delta Z_k}} \tag{4.38}$$

where indeed the result from (4.38) is of the same form as (4.21), i.e. it is a product of a Poisson component and MBM component. It can then be shown that the resulting non-PGFL equations are given by [32]

$$\nu_k(\boldsymbol{x}_k) = (1 - P_D(\boldsymbol{x}_k))\nu_{k|k-1}(\boldsymbol{x}_k) \tag{4.39}$$

for the Poisson component, while the equations for the MBM component are given by four cases, depending on the posterior hypothesis each track can participate in.

No target

In case a target does not exist in the hypothesis, we get that only w_k^{lt} is defined and $w_k^{lt} = 1$.

New target

If a measurement $\boldsymbol{z}_k^{lt}$ is declared as a new target with index t, it is initialized by the equations

$$w_k^{lt} = \lambda(\boldsymbol{z}_k^{lt}) + \int \nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k)\, \mathrm{d}\boldsymbol{x}_k \tag{4.40}$$

$$r_k^{lt} = \frac{\int \nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k)\, \mathrm{d}\boldsymbol{x}_k}{\lambda(\boldsymbol{z}_k^{lt}) + \int \nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k)\, \mathrm{d}\boldsymbol{x}_k}, \tag{4.41}$$

$$p_k^{lt}(\boldsymbol{x}_k) = \frac{\nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k)}{\int \nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k)\, \mathrm{d}\boldsymbol{x}_k}, \tag{4.42}$$

where $\boldsymbol{z}_k^{lt}$ denotes the measurement $\boldsymbol{z}_k^j$ that was associated to track t under hypothesis l.

Misdetection

In case we declare a target as misdetected we update its state space by

$$w_k^{lt} = w_{k|k-1}^{lt} \left(1 - r_{k|k-1}^{lt} + r_{k|k-1}^{lt} p_{k|k-1}^{lt}[1 - P_D(\boldsymbol{x}_k)]\right) \tag{4.43}$$

$$r_k^{lt} = \frac{r_{k|k-1}^{lt} p_{k|k-1}^{lt}[1 - P_D(\boldsymbol{x}_k)]}{1 - r_{k|k-1}^{lt} + r_{k|k-1}^{lt} p_{k|k-1}^{lt}[1 - P_D(\boldsymbol{x}_k)]} \tag{4.44}$$

$$p_k^{lt}(\boldsymbol{x}_k) = \frac{(1 - P_D(\boldsymbol{x}_k)) p_{k|k-1}^{lt}(\boldsymbol{x}_k)}{p_{k|k-1}^{lt}[1 - P_D(\boldsymbol{x}_k]} \tag{4.45}$$

Detection of existing target

Lastly, in case we declare the measurement $\boldsymbol{z}_k^{lt}$ a detection of an existing target $\boldsymbol{x}_k^t$, we perform the update

$$w_k^{lt} = w_{k|k-1}^{lt} r_{k|k-1}^{lt} \int p_{k|k-1}^{lt}(\boldsymbol{x}_k) p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k) \, \mathrm{d}\boldsymbol{x}_k \tag{4.46}$$

$$r_k^{lt} = 1 \tag{4.47}$$

$$p_k^{lt}(\boldsymbol{x}_k) = \frac{p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k) p_{k|k-1}^{lt}(\boldsymbol{x}_k)}{\int p_z(\boldsymbol{z}_k^{lt}|\boldsymbol{x}_k) P_D(\boldsymbol{x}_k) p_{k|k-1}^{lt}(\boldsymbol{x}_k) \, \mathrm{d}\boldsymbol{x}_k} \tag{4.48}$$

4.5 Model simplifications

The equations in Chapters 4.3 and 4.4 are highly abstract in nature, and to make them useful for implementation purposes some model simplifications are necessary. Specifically, we will focus on the track weights w_k^{lt} as they are required in Chapter 5 when we construct the multi-hypothesis factor graph.

We assume the misdetection probability $P_D(\boldsymbol{x}_k)$ to be constant, such that

$$P_D(\boldsymbol{x}_k) = P_D, \quad 0 \leq P_D \leq 1. \tag{4.49}$$

We will also assume a constant clutter intensity λ. Finally, recall that we use l^{jt} as symbol for the likelihood, which from (3.4) was given as

$$l^{jt} = \int p_k^t(\boldsymbol{x}_k^t|Z_{1:k-1}) p_z(\boldsymbol{z}_k^j|\boldsymbol{x}_k^t) \, \mathrm{d}\boldsymbol{x}_k^t$$

where we below instead use the notation l^{lt} for l^{jt} to indicate that the likelihood is constructed from the association between track t and measurement j in hypothesis l.

With these simplifications, the updated track weights for misdetection of existing tracks in (4.43) reduces to

$$w_k^{lt} = w_{k|k-1}^{lt} \left(1 - r_{k|k-1}^{lt} + r_{k|k-1}^{lt} p_{k|k-1}^{lt}[1 - P_D]\right) \tag{4.50}$$

$$= w_{k|k-1}^{lt} \left(1 - r_{k|k-1}^{lt} + r_{k|k-1}^{lt} (1 - P_D) p_{k|k-1}^{lt}[1]\right) \tag{4.51}$$

$$= w_{k|k-1}^{lt} \left(1 - r_{k|k-1}^{lt} P_D\right) \tag{4.52}$$

where we have used that

$$p^{lt}_{k|k-1}[1] = \int p^{lt}_{k|k-1}(\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{x}_k \tag{4.53}$$

$$= 1 \tag{4.54}$$

since $p^{lt}_{k|k-1}(\boldsymbol{x}_k)$ is a proper distribution. For detection we get

$$w^{lt}_k = w^{lt}_{k|k-1} r^{lt}_{k|k-1} \int p^{lt}_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}^{lt}_k|\boldsymbol{x}_k) P_D\,\mathrm{d}\boldsymbol{x}_k \tag{4.55}$$

$$= w^{lt}_{k|k-1} r^{lt}_{k|k-1} P_D \int p^{lt}_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}^{lt}_k|\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{x}_k \tag{4.56}$$

$$= w^{lt}_{k|k-1} r^{lt}_{k|k-1} P_D l^{lt}. \tag{4.57}$$

Unfortunately, the weights for new tracks are not as straight-forward to simplify. We first rewrite the weights to

$$w^{lt}_k = \lambda + \int \nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}^{lt}_k|\boldsymbol{x}_k) P_D\,\mathrm{d}\boldsymbol{x}_k \tag{4.58}$$

$$= \lambda + P_D \int \nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}^{lt}_k|\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{x}_k \tag{4.59}$$

and see that we are still required to compute $\int \nu_{k|k-1}(\boldsymbol{x}_k) p_z(\boldsymbol{z}^{lt}_k|\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{x}_k$ which is in general infeasible. A common assumption to make is that $\nu_{k|k-1}(\boldsymbol{x}_k)$ can be approximated as a *Gaussian mixture* over Gaussians linear in $\boldsymbol{x}_k$, such that the integral becomes a sum over solvable Gaussian integrals. This is done in e.g. [37]. Assuming this is the case, we call the result simply $\tilde{\nu}^{lt}_k$. Hence, the new-track weight becomes

$$w^{lt}_k = \lambda + P_D \tilde{\nu}^{lt}_k. \tag{4.60}$$

4.6 Recycling of tracks and conservation of track cardinality

A large bottleneck to multiple hypothesis tracking, and in particular in PMBM, is managing the ever-growing number of hypotheses and the resulting number of tracks. The following section will discuss how to mitigate this problem by pruning tracks by *recycling* and keeping the filter consistent by arguing about *track cardinality balancing*.

4.6.1 Enumerating the M best hypotheses with Murty's method

Before going into the problem of conservation of track cardinality, we first set the stage by presenting the same introduction to *Murty's method* as in the preceding project report. 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 [38]. The algorithm that makes this possible is called *Murty's method*, named after its inventor [39] which published the method back in 1968. The method was later adopted into the MTT community by Cox, Miller et. al in [40] which optimized the algorithm for use in MHT. Later, by Danchick and Newnam in [41], 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 3.1.1. Common choices [18] are the *Hungarian method* [42], the *auction method* [43] and the *Jonker-Volgenant (JV) algorithm* [44]. In [40] 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 [18].

A multi-cluster, multi-hypothesis generalization of Murty's method based on the *branch-and-bound* optimization method [45] is described in [18], where the notion of multi-cluster was introduced in Chapter 3.6. In particular, one of the main strengths with this implementation is that it allows for efficient, approximate hypothesis enumeration even in the presence of cluster merging.

4.6.2 Preserving track cardinality with track recycling

Even though we only keep the M best hypotheses after every time step, we are still interested in approximating the association marginals for the posterior tracks for the following reason. When pruning hypotheses after every time step, we want to keep our filter *consistent*. For the purpose of this text, it suffice to say that a filter is consistent if *the output of the filter on average describes its errors well* [33]. Intuitively, one can think of this as the filter "knowing" if its correctly estimating what it tries to estimate or not and assigning a reasonable uncertainty to this estimate. A more rigorous introduction to filter consistency can be found in [34]. In PMBM, one way to improve filter consistency is to conserve the *expected track cardinality*. We illustrate what we mean by first decomposing the PMBM posterior mixture in some timestep k over L posterior hypotheses into the

posterior hypotheses from Murty's method, denoted $\mathscr{M}_k$, and the remaining hypotheses, denoted $\mathscr{R}_k$,

$$G_k^{\text{mbm}}[h] = \sum_{l=1}^{L} w_k^l G_k^{\text{mb}}[h] \tag{4.61}$$

$$= \sum_{l \in \mathscr{M}_k} w_k^l G_k^{\text{mb}}[h] + \sum_{l \in \mathscr{R}_k} w_k^l G_k^{\text{mb}}[h] \tag{4.62}$$

where we must have that $\mathscr{M}_k \cup \mathscr{R}_k = \theta_{1:k}$ and $\mathscr{M} \cap \mathscr{R}_k = \emptyset$, i.e. that the sets of hypotheses $\mathscr{M}_k$ and $\mathscr{R}_k$ make up all the posterior hypotheses $\theta_{1:k}$ exactly and disjointly. We then calculate the PHD of the distribution, which is defined as [9]

$$\beta(\boldsymbol{x}) = \frac{\delta G}{\delta \boldsymbol{x}}[1] \tag{4.63}$$

and has the property that its integral over any region $S \subseteq \mathbb{R}^d$ with $\boldsymbol{x} \in \mathbb{R}^d$ is equal to the expected number of targets in S. Using the fact that the PHD is indeed linear from (4.63), we get the result that

$$\beta^{\text{mbm}}(\boldsymbol{x}) = \sum_{l \in \mathscr{M}_k} w_k^l \beta^{\text{mb}}(\boldsymbol{x}) + \sum_{l \in \mathscr{R}_k} w_k^l \beta^{\text{mb}}(\boldsymbol{x}) \tag{4.64}$$

which we can interpret as follows. The functions $\beta^{\text{mb}}(\boldsymbol{x})$ effectively denote the expected number of targets under the hypothesis l, which we weigh by w_k^l. Thus, in order to keep expected track cardinality, at least approximately, when using Murty's, we need the partial sum over remaining hypotheses to get negligible. Usually, this happens because Murty's finds the M best hypotheses, i.e. with the largest weights w_k^l, and so the partial sum over hypotheses from Murty's dominates the other partial sum when the hypothesis distribution is sufficiently peaked.

However, should the total number of posterior hypotheses be very large, such that $L >> M$, and the corresponding hypothesis distribution more flat, then we end up pruning a non-negligible amount of track mass, effectively removing relevant information from the system. A practical scenario where this can be a problem is where we have a lot of targets densely packed, little clutter and association ambiguity. Initially, we are inclined to initialize tracks with high confidence to be from true targets. This makes the number of tracks and hypotheses grow quickly, and we eventually start pruning hypotheses and tracks when using Murty's. Deleting track cardinality this way makes the filter believe there are less targets present than there really are, making it more hesitant to initialize tracks with possibly catastrophic consequences.

A more favorable approach would therefore be to instead actively prune hypotheses and tracks in a way that let the filter still quickly initialize new tracks confidently in such an environment. We call this concept *aggressive recycling*. The idea of track recycling as a way of achieving cardinality balance was first described by Williams in [46] for the *Poisson multi-Bernoulli* (PMB) filter, the single-hypothesis relative of the PMBM filter. They propose to recycle Bernoulli components of low quality, i.e. tracks with either low existence probability or large covariance, by inserting them back into the Poisson component as a pruning technique that reuses information. This hinges on the fact that any point process can be approximated with a best-fit Poisson process by using its PHD as the intensity [9]. We can translate this method to the multi-hypothesis case by using the recycled Poisson component of track t with Poisson intensity

$$\lambda^t(\boldsymbol{x}_k) = q_k^t p_k^t(\boldsymbol{x}_k^t | Z_{1:k}) \tag{4.65}$$

where $p_k^t(\boldsymbol{x}_k^t|Z_{1:k})$ is the posterior state distribution to track t and q_k^t is the *total track probability* [47], defined as

$$q_k^t = r_k^t \sum_{\theta_{1:k} \,:\, t \in \theta_{1:k}} \mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\} \tag{4.66}$$

$$= r_k^t \pi_k^t \tag{4.67}$$

where r^t is the existence probability of track t, $\sum_{\theta_{1:k} \,:\, t \in \theta_{1:k}}$ denotes the sum over all posterior hypotheses $\theta_{1:k}$ containing track t and

$$\pi_k^t = \sum_{\theta_{1:k} \,:\, t \in \theta_{1:k}} \mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\} \tag{4.68}$$

is a temporary notation for the *association marginal* to track t, a quantity we will become all too familiar with in the sequel.

Let us for the time being assume we have access to the exact marginals π^t from some oracle. We are still interested in doing track recycling for real-time estimation as it is computationally cheaper to keep tracks in the Poisson component rather than the MBM component. The exact details are outside the scope of this thesis, but the main idea is that data association is significantly cheaper for tracks in the Poisson component as they are propagated using the PHD filter which avoids explicit hypothesis enumeration [9]. We recall from (4.15) that the hypothesis posterior distribution is given up to scale from

the hypothesis weights w_k^l,

$$\begin{aligned}\mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\} &= \frac{1}{Z} w_k^l \\ &\propto w_k^l.\end{aligned}$$

We also assume we have access to the exact normalization constant Z from some oracle. Track recycling can then be done as follows. The total track mass a track contributes with is given from the marginal π^t, which involves a sum over all hypotheses. Rewriting (4.68) into the partial sums over Murty's hypotheses $\mathscr{M}_k$ and the remaining $\mathscr{R}_k$ gives

$$\begin{aligned}\pi^t &= \sum_{\theta_{1:k} \,:\, t \in \theta_{1:k}} \mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\} && (4.69)\\ &= \sum_{\theta_{1:k} \,:\, t \in \mathscr{M}_k} \mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\} + \sum_{\theta_{1:k} \,:\, t \in \mathscr{R}_k} \mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\} && (4.70)\\ &= \sum_{\theta_{1:k} \,:\, t \in \mathscr{M}_k} \frac{w_k^l}{Z} + \sum_{\theta_{1:k} \,:\, t \in \mathscr{R}_k} \mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\} && (4.71)\\ &= \sum_{\theta_{1:k} \,:\, t \in \mathscr{M}_k} \frac{w_k^l}{Z} + \pi_R^t && (4.72)\end{aligned}$$

where the weights w_k^l in $\sum_{\theta_{1:k} \,:\, t \in \mathscr{M}_k} w_k^l / Z$ are found by Murty's and we define the remaining marginal $\pi_R^t = \sum_{\theta_{1:k} \,:\, t \in \mathscr{R}_k} \mathsf{Pr}\{\theta_{1:k} \mid Z_{1:k}\}$. The track mass enumerated from Murty's is put in the MBM component. Thus, we use the remaining marginal π_R^t in the total track probability in (4.67) for the recycled Poisson component to maintain track cardinality balance, which we compute from

$$\pi_R^t = \pi_k^t - \sum_{\theta_{1:k} \,:\, t \in \mathscr{M}_k} \frac{w_k^l}{Z}. \tag{4.73}$$

We see by looking at (4.73) that we can maintain exact track cardinality balance by having access to the association marginal π_k^t and normalization constant Z, which we in general do not. This motivates us to develop methods for efficiently estimating them, which we do in Part II.

II

Multi-cluster, multi-hypothesis association methods

5 | Constructing the association factor graph

In [1] we developed two algorithms for calculating marginal track probabilities in multiple hypothesis tracking. Here, we generalize these to a multi-cluster scenario. Furthermore, we present novel equations for estimating the normalization constant of the joint association posterior by the *Bethe pseudodual function*. A copy of the article [1] is available in Appendix C

The following section will review the multi-hypothesis factor graph presented in [1] through the lens of PMBM and also generalize it to the multi-cluster scenario.

5.1 Deriving the joint association posterior

To start, we will use the fact that the joint association posterior $\mathsf{Pr}\{\theta_{1:k}|Z_{1:k}\}$ evaluated in posterior hypothesis $\theta^l_{1:k}$ can be written as proportional to the product over the track weights of tracks that exist in the hypothesis,

$$\mathsf{Pr}\{\theta^l_{1:k}|Z_{1:k}\} \propto w^l_k = \prod_{t\in\theta^l_{1:k}} w^{lt}_k. \tag{5.1}$$

Now, since the tracks existing in hypothesis $\theta^l_{1:k}$ either are new tracks or existing tracks that are misdetected or detected, the product in (5.1) can be written as

$$\prod_{t=1}^{n} w^{lt}_k = \prod_{t\in B^l_k} w^{lt}_k \prod_{t\in M^l_k} w^{lt}_k \prod_{t\in D^l_k} w^{lt}_k \tag{5.2}$$

where D^l_k denotes the set of existing tracks that were detected, M^l_k the set of existing tracks that were misdetected and B^l_k the set of new tracks.

Combining this with the recursive definition of the weights, we arrive at

$$\begin{aligned}\mathsf{Pr}\{\theta^l_{1:k}|Z_{1:k}\} &\propto \prod_{t\in\theta^l} w^{lt}_k \qquad (5.3)\\ &= \underbrace{\prod_{t\in B^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right)}_{\text{New tracks}} \cdot \underbrace{\prod_{t\in M^l_k} w^{lt}_{k|k-1}\left(1 - r^{lt}_{k|k-1}P_D\right)}_{\text{Misdetection}} \\ &\cdot \underbrace{\prod_{t\in D^l_k} w^{lt}_{k|k-1} r^{lt}_{k|k-1} P_D l^{lt}}_{\text{Detection}}. \qquad (5.4)\end{aligned}$$

We now collect the prior weights $w^{lt}_{k|k-1}$ from the detected and misdetected tracks products into its own product $\prod_{t\in D^l_k\cup M^l_k} w^{lt}_{k|k-1}$ which we recognize as proportional to the parent hypothesis probability $\mathsf{Pr}\{\theta^p_{1:k-1} \mid Z_{1:k-1}\}$ since we must have that

$$\theta^p_{1:k-1} = D^l_k \cup M^l_k \qquad (5.5)$$

exactly since $D^l_k \cup M^l_k$ is the set of existing tracks from $\theta^p_{1:k-1}$ and

$$w^{lt}_{k|k-1} = w^{lt}_{k-1}. \qquad (5.6)$$

In total, this means that the joint association posterior can be written as

$$\begin{aligned}\mathsf{Pr}\{\theta^r_{1:k}|Z_{1:k}\} \propto \mathsf{Pr}\{\theta^l_{1:k-1}|Z_{1:k-1}\} &\prod_{t\in B^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right) \prod_{t\in M^l_k}\left(1 - r^{lt}_{k|k-1}P_D\right) \\ \cdot \prod_{t\in D^l_k} r^{lt}_{k|k-1} P_D l^{lt}&. \qquad (5.7)\end{aligned}$$

We make one final trick to get (5.7) on a common form. We multiply by 1 in the following way

$$\begin{aligned}\prod_{t\in B^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right) &= \prod_{t\in B^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right) \cdot \frac{\prod_{t\in D^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right)}{\prod_{t\in D^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right)} \qquad (5.8)\\ &= \prod_{t\in B^l_k\cup D^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right) \cdot \frac{1}{\prod_{t\in D^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right)} \qquad (5.9)\\ &\propto \frac{1}{\prod_{t\in D^l_k}\left(\lambda + P_D\tilde{\nu}^{lt}_k\right)} \qquad (5.10)\end{aligned}$$

where we move the $\prod_{t\in B_k^l \cup D_k^l}$ product into the proportionality sign in (5.10) since the product is over the factors $\lambda + P_D \tilde{\nu}_k^{lt}$ for each measurement in Z_k, which is constant and the same for all hypotheses. Thus, the joint association posterior that is used is

$$\mathsf{Pr}\{\theta_{1:k}^r | Z_{1:k}\} \propto \mathsf{Pr}\{\theta_{1:k-1}^l | Z_{1:k-1}\} \prod_{t\in M_k^l} \left(1 - r_{k|k-1}^{lt} P_D\right) \prod_{t\in D_k^l} \frac{r_{k|k-1}^{lt} P_D l^{lt}}{\lambda + P_D \tilde{\nu}_k^{lt}}. \tag{5.11}$$

5.2 Single-cluster multi-hypothesis factor graph

We here proceed by showing how we can rewrite the joint single-cluster, multi-hypothesis association distribution in (5.11) in an overparameterized form for use in a factor graph as is done in [8] originally for single-hypothesis and described in [1] for multi-hypothesis purposes. Firstly, we drop all references to a particular timestep k and sequences in time $1{:}k{-}1$ and $1{:}k$ for notational simplicity. We denote the prior hypothesis variable by θ as seen previously, only we now drop the reference to a particular parent hypothesis. The posterior hypothesis will be denoted by $\theta_{1:k}$ to distinguish it from the prior hypothesis θ. The prior distribution will in the sequel be referred to by $\varphi(\theta)$ such that

$$\varphi(\theta) = \mathsf{Pr}\{\theta_{1:k-1} \mid Z_{1:k-1}\}. \tag{5.12}$$

We introduce a^t as the *track association variable* to indicate the measurement index that track t is associated with, where $a^t = 0$ indicates misdetection, $a_k^t = j$, $j \in \{1, \ldots, m\}$ indicates detection by measurement j out of m_k measurements and $a^t = N$ indicates *nonexistence*. Nonexistence is necessary as we can only declare tracks as misdetected or detected for hypotheses where the track exists, and so we need to be able to assign probability to the event that no assignment is possible. To enforce the constraint that a track a^t is only declared misdetected or detected when considering a parent hypothesis θ where it exists, and otherwise nonexistent, we use the consistency factor $\zeta^t(\theta, a^t)$ defined as

$$\zeta^t(\theta, a^t) = \begin{cases} 1, & \begin{array}{c} t \in \theta \wedge a_k^t \neq N \\ \vee\, t \notin \theta \wedge a_k^t = N \end{array} \\ 0, & \text{otherwise} \end{cases} \tag{5.13}$$

where the expressions $t \in \theta$ and $t \notin \theta$ are used as a logical statements to indicate true if $t \in \theta$ and false if $t \notin \theta$, respectively. We also need to consider the products over the associations that can be made for each track. To achieve this we can define the unary

factor $\psi^t(a^t)$ for each track $t = 1, \dots, n$ such that we get

$$\psi^t(a^t = 0) = 1 - r^t P_D, \tag{5.14a}$$

$$\psi^t(a^t = j) = \frac{r^t P_D l^{jt}}{\lambda + P_D \tilde{\nu}^{jt}} \tag{5.14b}$$

$$\psi^t(a^t = N) = 1 \tag{5.14c}$$

to be consistent with (5.11). The definition in (5.14c) follows from $w_k^{lt} = 1$ for tracks that do not exist in the hypothesis θ^l and is conveniently reintroduced here. Although the parameterization above is sufficient, we overparameterize the distribution by introducing the *measurement association variables* b^j for each measurement j. We let $b^j \in \{0, 1, \dots, n\}$ where $b^j = 0$ indicates false alarm and $b^j = t$, $t \in \{1, \dots, n\}$ indicates that measurement j is a detection of track t. In [8] they argue that this is useful to ensure that LBP will have a unique fixed point, as discussed in Chapter 2.3.4. In order to assign 0 probability to invalid assignments, i.e. where a track and measurement are not associated to each other simultaneously, equivalently the at-most-one assumption mentioned in Chapter 3.1.1, we introduce the consistency factor $\gamma^{jt}(a^t, b^j)$ with definition

$$\gamma^{jt}(a^t, b^j) = \begin{cases} 0, & \begin{array}{l} a^t = j \wedge b^j \neq t \\ \vee\, a^t \neq j \wedge b^j = t\,. \end{array} \\ 1, & \text{otherwise} \end{cases} \tag{5.15}$$

With the factors introduced in (5.12) to (5.15), we rewrite (5.11) as follows. Under the consistency factors ζ^t and γ^{jt} we can safely consider all tracks and measurements collectively. In the first step we only consider tracks to get

$$\mathsf{Pr}\{\theta_{1:k}|Z_{1:k}\} \propto \underbrace{\mathsf{Pr}\{\theta_{1:k-1}|Z_{1:k-1}\}}_{\varphi(\theta)} \underbrace{\prod_{t \in M_k^l} \left(1 - r^t P_D\right) \prod_{t \in D_k^l} \frac{r^t P_D l^{lt}}{\lambda + P_D \tilde{\nu}^{lt}}}_{\prod_{t \in \theta} \zeta^t(\theta, a^t) \psi^t(a^t)} \tag{5.16}$$

$$= \varphi(\theta) \prod_{t \in \theta} \zeta^t(\theta, a^t) \psi^t(a^t) \tag{5.17}$$

$$= \varphi(\theta) \prod_{t \in \theta} \zeta^t(\theta, a^t) \psi^t(a^t) \prod_{t \notin \theta} \zeta^t(\theta, a^t) \psi^t(a^t) \tag{5.18}$$

$$= \varphi(\theta) \prod_{t=1}^{n} \zeta^t(\theta, a^t) \psi^t(a^t) \tag{5.19}$$

where we used that $\prod_{t \notin \theta} \zeta^t(\theta, a^t)\psi^t(a^t) = 1$ since $\psi^t(a^t = N) = 1$ and $\zeta^t(\theta, a^t = N) = 1$ for $t \notin \theta$. We then add a product over all b^j for each track a^t to finally arrive at

$$\mathsf{Pr}\{\theta_{1:k}|Z_{1:k}\} \propto \varphi(\theta) \prod_{t=1}^{n} \left(\zeta^t(\theta, a^t)\psi^t(a^t) \prod_{j=1}^{m} \gamma^{jt}(a^t, b^j) \right). \tag{5.20}$$

An illustration how such a factor graph can look like can be found in Figure 5.1, which is the same example as in the preceding project report.

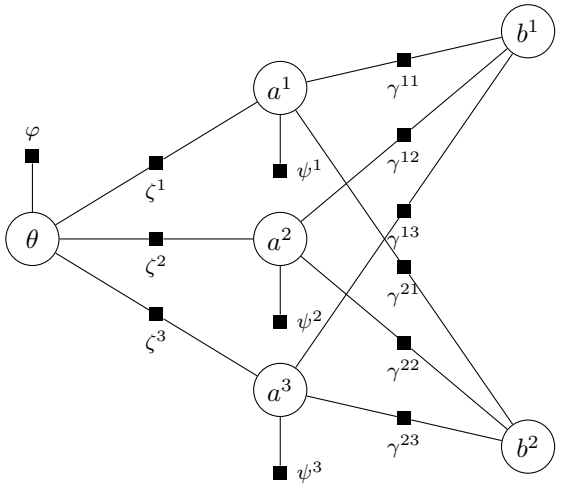


Figure 5.1: *An illustrative example of a multi-hypothesis association factor graph with three tracks a^1, a^2 and a^3 and two measurements b^1 and b^2.*

5.3 Generalizing to multiple clusters

The multi-hypothesis association posterior in (5.20) is for a single cluster. Given the way we define clusters in Chapter 3.6, generalizing the distribution to multiple clusters is trivial as we assume that clusters are independent, meaning the multi-cluster posterior is nothing more than a product over the single-cluster posteriors

$$\mathsf{Pr}\{\Theta_{1:k}|Z_{1:k}\} = \prod_{c=1}^{C} \mathsf{Pr}\{\theta^c_{1:k}|Z_{1:k}\}. \tag{5.21}$$

where $\Theta_{1:k}$ denotes the joint set over all prior hypotheses $\theta^c_{1:k}$ and the superscript c here indicates that $\theta^c_{1:k}$ belongs to cluster c. An example of how such a multi-cluster, multi-hypothesis factor graph could look like can be found in Figure 5.2

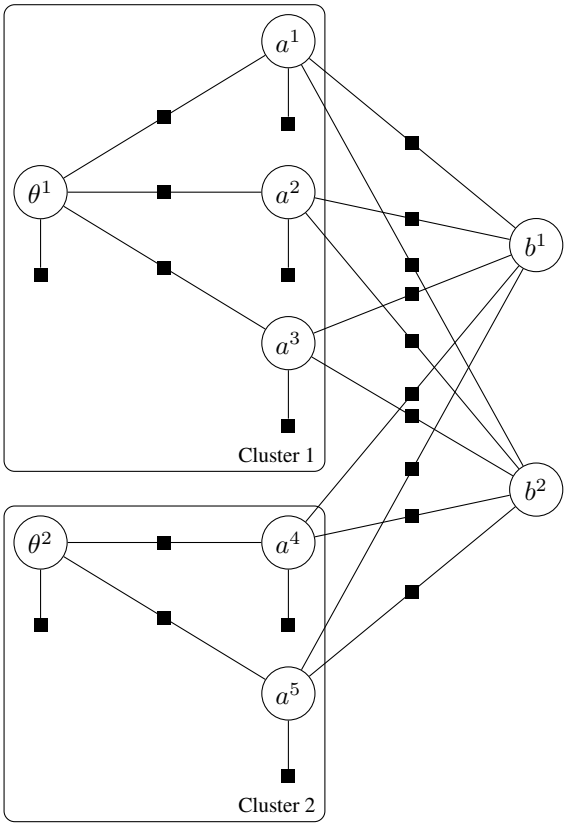


Figure 5.2: *Multi-cluster factor graph of test case with two clusters, five tracks and two measurements.*

6 | Marginals and normalization constant by LBP

6.1 Multi-cluster, multi-hypothesis LBP

In the preceding project report, novel LBP messages were derived to compute approximate association marginals on a single-cluster, multi-hypothesis factor graph with factorization given in (5.20). We repeat the argumentation made in the preceding project report here. The work was based upon [8] where the authors made the observation that the LBP messages in a single-hypothesis, single-cluster association graph have a particular structure to them which allows for clever normalizations that reduce computation complexity and yields simpler expressions. This holds for the multi-hypothesis association graph as 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.

In this thesis, we generalize the LBP message equations to the novel *multi-cluster* messages. A large motivating factor for doing this is to be able to compute association marginal approximations even in the presence of cluster merging, which we briefly discussed in Chapter 3.6.1.

We use four different types of messages. 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^c to a track t_c, both in cluster c, is denoted by σ_{t_c} and finally, the message from a track t to the prior hypothesis θ^c in cluster c is denoted by ρ_t. Note that we only need to explicitly state what cluster c a track t_c belongs to for the hypthesis-to-track message σ_{t_c}, as will be made clear below. The message definitions are summarized in Table A.1 and their directions illustrated in

Figure A.1. The resulting Theorem with proof can be found in Theorem 1.

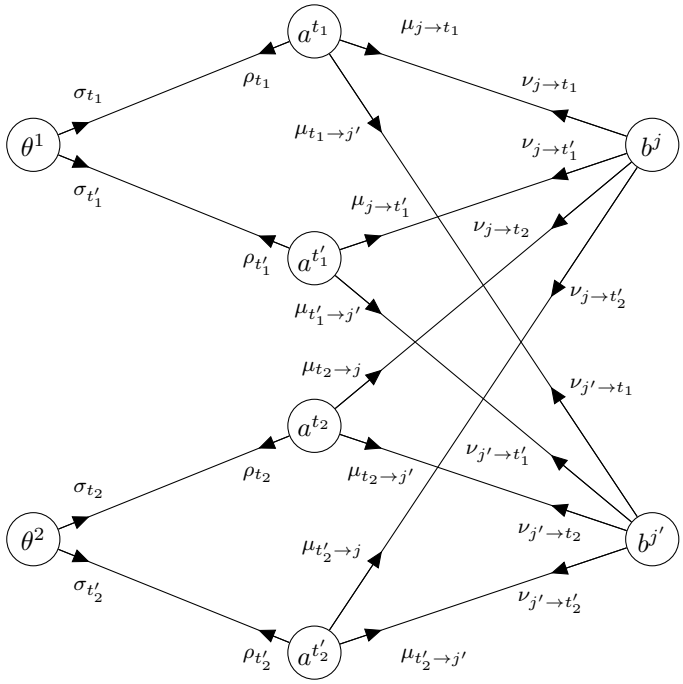


Figure 6.1: *Message direction example for multi-cluster scenario with two clusters.*

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_c}	$\theta^c \to a^{t_c}$
Track-to-hypothesis	ρ_t	$a^t \to \theta^c$

Table 6.1: *Message types in multi-cluster, multi-hypothesis association graph.*

Theorem 1 (The message definitions for multi-cluster, multi-hypothesis LBP)**.** *Given an association graph of the same structure as in Figure 5.2 where the factors are defined as in* (5.12) *to* (5.15)*, the normalized messages used in multi-cluster, multi-hypothesis LBP are given as*

$$\mu_{t\to j} = \frac{\psi^t(j)}{\psi^t(0) + \sum_{j'\neq j, j'>0} \psi^t(j')\nu_{j'\to t} + \sigma_t}, \tag{6.1a}$$

$$\nu_{j\to t} = \frac{1}{1 + \sum_{t'\neq t, t'>0} \mu_{t'\to j}}, \tag{6.1b}$$

$$\sigma_{t_c} = \rho_{t_c} \cdot \frac{\sum\limits_{\theta^c \,:\, t_c \notin \theta^c} \varphi(\theta^c) \prod\limits_{t'_c \in \theta} \rho_{t'_c}}{\sum\limits_{\theta^c \,:\, t_c \in \theta^c} \varphi(\theta^c) \prod\limits_{t'_c \in \theta^c} \rho_{t'_c}}, \tag{6.1c}$$

$$\rho_t = \psi^t(0) + \sum_{j=1}^{m_k} \psi^t(j)\nu_{j\to t} \tag{6.1d}$$

where $\sum_{j'\neq j, j'>0}$ denotes the sum over all values $j' = 1, \ldots, m_k$ except for j for m_k measurements, $\sum_{t'\neq t, t'>0}$ denotes the sum over all values $t' = 1, \ldots, n_k$ except for t for n_k tracks, $\sum_{\theta^c \,:\, t_c \in \theta^c}$ denotes the sum over all prior hypotheses θ^c in cluster c where track t_c exists and vice versa for $\sum_{\theta^c \,:\, t_c \notin \theta^c}$ and $\prod_{t'_c \in \theta^c}$ denotes the product over all tracks t_c that exist in the prior hypothesis θ^c in cluster c.

Proof. See Appendix A. ■

Assuming the LBP algorithm converges, we can compute the approximate association marginals from

$$\hat{p}(a^t|Z_{1:k}) \propto \begin{cases} \psi^t(0), & a^t = 0 \\ \psi^t(j)\nu_{j\to t}, & a^t = 1, \ldots, m_k \\ \sigma_t, & a^t = N \end{cases} \tag{6.2}$$

where we in σ_t drop the indication of cluster since it is unambiguous given a track t. The measurement marginals can be computed with

$$\hat{p}(b^j|Z_{1:k}) \propto \begin{cases} 1, & b^j = 0, \\ \mu_{t\to j}, & b^j = 1, \ldots, n_k \end{cases} \tag{6.3}$$

and the prior hypothesis posterior with

$$\hat{p}(\theta|Z_{1:k}) \propto \varphi(\theta) \prod_{t \in \theta} \rho_t \tag{6.4}$$

6.2 Normalization constant estimation by Bethe approximation

As an alternative approach to full LBP in [1], a hypothesis-conditioned approach was also proposed in [1] that allowed for using the LBP in [8] which has been proved to have many desirable properties discussed in Chapter 2.3.4. This approach, however, depended on estimating the *hypothesis-conditioned likelihood* $p(Z_k|Z_{1:k-1}, \theta)$. In [1] the likelihood was approximated using FISST and approximating the Binomial distribution over measurements as a Poisson distribution, similarly to what is done in the PHD filter [9]. The results showed that this gave acceptable performance, but that the accuracy of the likelihood was the main reason for inaccuracies.

In the present work we therefore derive novel equations for computing the Bethe constant based on the Bethe pseudodual, discussed in Chapter 2.3.3. We motivate using the Bethe pseudodual for two reasons. The first is, as already mentioned, that the pseudodual evaluates exactly to Bethe free energy function at the fixed point of LBP. The other is implementation-wise, as this provides us with sensible metric to measure convergence when using LBP as in [26]. In particular, however, we can also use it with the messages in a multi-hypothesis factor graph, which is useful as both [26] and [32] only present convergence metrics for single-cluster, single-hypothesis factor graphs.

In Theorem 2 we prove the pseudodual of the single-cluster, multi-hypothesis association distribution. In practice, however, we are more interested in the multi-cluster case. A particular structure will become evident from the single-cluster case which makes generalizing to multi-cluster straight-forward. Concerning ourselves with the single-cluster case makes the proof slightly less involved. Additionally, we present the Bethe pseudodual for the single-cluster, single-hypothesis case to estimate the same hypothesis-conditioned likelihood as discussed above.

Theorem 2 (The pseudodual of the Bethe free energy function for the single-cluster, multi-hypothesis association graph)**.** *Given the factors of the single-cluster, multi-hypothesis association graph and the LBP messages in [1], the pseudodual of the corresponding Bethe free energy function is given as*

$$F_B^{\#} = (n-1)\ln Z_\theta + m\sum_{t=1}^{n}\ln Z_t + (n-1)\sum_{j=1}^{m}\ln Z_j - \sum_{t=1}^{n}\ln Z_{t\theta} - \sum_{t=1}^{n}\sum_{j=1}^{m}\ln Z_{tj} \tag{6.5}$$

where n denotes the number of targets t, m the number of measurements j and Z_θ, Z_t, Z_j, $Z_{t\theta}$ and Z_{tj} are the LBP belief normalization constants and can be computed with the equations

$$Z_\theta = \sum_{\theta}\varphi(\theta)\prod_{t\in\theta}\rho_t, \tag{6.6}$$

$$Z_t = \psi^t(0) + \sum_{j=1}^{m}\psi^t(j)\nu_{j\to t} + \sigma_t, \tag{6.7}$$

$$Z_j = 1 + \sum_{t=1}^{n}\mu_{t\to j}, \tag{6.8}$$

$$Z_{t\theta} = \frac{\psi^t(0) + \sum_{j=1}^{m}\psi^t(j)\nu_{j\to t}}{\rho_t}\sum_{\theta:t\in\theta}\varphi(\theta)\prod_{t'\in\theta}\rho_{t'} + \sum_{\theta:t\notin\theta}\varphi(\theta)\prod_{t'\in\theta}\rho_{t'}, \tag{6.9}$$

$$Z_{tj} = \left(1 + \sum_{\substack{t'=1\\t'\neq t}}^{n}\mu_{t'\to j}\right)\left(\psi^t(0) + \sum_{\substack{j'=1\\j'\neq j}}^{m}\psi^t(j')\nu_{j'\to t} + \sigma_t\right) + \psi^t(j) \tag{6.10}$$

Proof. See Appendix B. ■

Based on Theorem 2 we present two corollaries for the Bethe pseudodual in the multicluster case and the single-cluster, single-hypothesis case in Corollary 1 and Corollary 2, respectively.

Corollary 1 (The pseudodual for multi-cluster, multi-hypothesis)**.** *From the result in Theorem 2 it should be reasonably clear that we can generalize the result to multi-cluster by stating the pseudodual as*

$$F^{\#}_{B,MC} = \sum_{c=1}^{C}(n_c - 1)\ln Z_{\theta^c} + m\sum_{t=1}^{n}\ln Z_t + (n-1)\sum_{j=1}^{m}\ln Z_j - \sum_{l=1}^{L}\sum_{t_c=1}^{n_c}\ln Z_{t_l\theta^l} - \sum_{t=1}^{n}\sum_{j=1}^{m}\ln Z_{tj}, \tag{6.11}$$

where θ^c denotes the hypothesis variable of cluster c with in total C clusters, t_c is a track in cluster c and n_c is the number of tracks in cluster c such that $\sum_{c=1}^{C} n_c = n$ and the normalization constant Z_{θ^c} is given as

$$Z_{\theta^c} = \sum_{\theta^c}\varphi(\theta^c)\prod_{t_c\in\theta^c}\rho_{t_c} \tag{6.12}$$

and the edge normalization constant $Z_{t_c\theta^c}$ is given as

$$Z_{t_c\theta^c} = \frac{\psi^{t_c}(0) + \sum_{j=1}^{m}\psi^{t_c}(j)\nu_{j\to t_c}}{\rho_{t_c}}\sum_{\theta^c:t_c\in\theta^c}\varphi(\theta^c)\prod_{t'_c\in\theta^c}\rho_{t'_c} + \sum_{\theta^c:t_c\notin\theta^c}\varphi(\theta^c)\prod_{t'_c\in\theta^c}\rho_{t'_c}. \tag{6.13}$$

Corollary 2 (The pseudodual for single-cluster, single-hypothesis)**.** *When we have an association case as in [8], the pseudodual takes the form*

$$F^{\#}_{B,SH} = (m-1)\sum_{t=1}^{n}\ln Z_t + (n-1)\sum_{j=1}^{m}\ln Z_j - \sum_{t=1}^{n}\sum_{j=1}^{m}\ln Z_{tj}, \tag{6.14}$$

Additionally, we normalize $\psi^t(a^t)$ by $\psi^t(0)$ and do not have any σ_t or ρ_t messages, so we use the equations

$$Z_t = 1 + \sum_{j=1}^{m}\psi^t(j)\nu_{j\to t}, \tag{6.15}$$

$$Z_j = 1 + \sum_{t=1}^{n}\mu_{t\to j}, \tag{6.16}$$

$$Z_{tj} = \left(1 + \sum_{\substack{t'=1\\t'\neq t}}^{n}\mu_{t'\to j}\right)\left(1 + \sum_{\substack{j'=1\\j'\neq j}}^{m}\psi^t(j')\nu_{j'\to t}\right) + \psi^t(j). \tag{6.17}$$

To build confidence that the above expressions are correct, we will test them on a simple single-cluster, single-hypothesis case with two tracks a^1 and a^2 that both gate a single measurement b^1. See Figure 6.2 for reference.

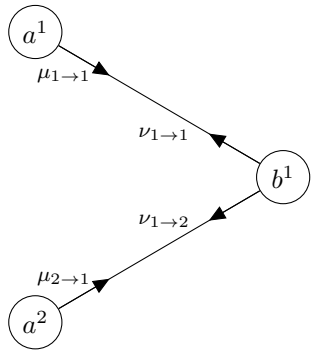


Figure 6.2: *Very simple single-cluster, single-hypothesis association case with two tracks and one measurement. The messages $\mu_{1\to1}$, $\mu_{2\to1}$, $\nu_{1\to1}$ and $\nu_{1\to1}$ are also indicated.*

In this case, the association graph is a tree, and so we expect the solution from LBP to be exact. To verify this, let us compute the exact normalization constant by hypothesis enumeration and compare it with the value we get with LBP and the Bethe pseudodual in Corollary 2.

Let $\psi^1(a^1)$ and $\psi^2(a^2)$ denote the unary factors for a^1 and a^2, respectively. Since this is a single-hypothesis case, both tracks have to exist, and so the nonexistence state $a^t = N$ vanishes. Additionally, we need to do the same as in [8] and normalize the unary factors by the misdetection probability, i.e. $\psi^1(a^1 = 0) = \psi^2(a^2 = 0) = 1$, such that the expressions in Corollary 2 remain correct. Since the unary factors therefore either are $\psi^t(a^t = 0) = 1$ or $\psi^t(a^t = 1)$, we use the short-hand notation $\psi^t = \psi^t(a^t = 1)$ for simplicity.

We have three association hypotheses, which are that both tracks are misdetected or that exactly one of them is detected by b^1. Thus, the exact normalization constant is

$$Z = f(a^1 = 0, a^2 = 0) + f(a^1 = 1, a^2 = 0) + f(a^1 = 0, a^2 = 1) \tag{6.18}$$
$$= 1 \cdot 1 + \psi^1 \cdot 1 + 1 \cdot \psi^2 \tag{6.19}$$
$$= 1 + \psi^1 + \psi^2 \tag{6.20}$$

where $f(a^1, a^2)$ is the underlying function of the association graph.

Let us now do the same with LBP. First, we must have that $\sigma_1 = \sigma_2 = 0$, which we

see from (6.1c) as the sum in the numerator is over all hypotheses that do not contain the track, which there are none of since this is single-hypothesis. We therefore get the empty sum which equals 0 by convention. This is also consistent with (6.2) as $\hat{p}(a^t = N|Z_{1:k}) = 0$ only for $\sigma_t = 0$, which we expect as the nonexistence association event should have probability 0. Secondly, since we only have one measurement, the sum $\sum_{j' \neq j, j' > 0} \psi^t(j')\nu_{j' \to t} = 0$ again since it becomes the empty sum. Thus, the track-to-measurement messages $\mu_{1\to 1}$ and $\mu_{2\to 1}$ become

$$\mu_{1\to 1} = \psi^1, \tag{6.21}$$

$$\mu_{2\to 1} = \psi^2. \tag{6.22}$$

Since $m = 1$, the sum $(m-1)\sum_{t=1}^{n} \ln Z_t = 0$. The second sum in (6.14) becomes

$$\begin{aligned}(n-1)\sum_{j=1}^{m} \ln Z_j &= \ln\left(1 + \sum_{t=1}^{n} \mu_{t\to j}\right) \\ &= \ln(1 + \mu_{1\to 1} + \mu_{2\to 1}) \\ &= \ln(1 + \psi^1 + \psi^2)\end{aligned} \tag{6.23}$$

where we substituted Z_j with (6.16) and inserted (6.21) and (6.22) for the messages $\mu_{1\to 1}$ and $\mu_{2\to 1}$. Lastly, the edge sum becomes

$$\begin{aligned}\sum_{t=1}^{n}\sum_{j=1}^{m} \ln Z_{tj} &= \sum_{t=1}^{n}\sum_{j=1}^{m} \ln\left[\left(1 + \sum_{\substack{t'=1 \\ t' \neq t}}^{n} \mu_{t'\to j}\right)\left(1 + \sum_{\substack{j'=1 \\ j' \neq j}}^{m} \psi^t(j')\nu_{j'\to t}\right) + \psi^t(j)\right] \\ &= \ln\left[(1 + \mu_{2\to 1})(1+0) + \psi^1\right] + \ln\left[(1+\mu_{1\to 1})(1+0) + \psi^2\right] \\ &= \ln(1 + \mu_{2\to 1} + \psi^1) + \ln(1 + \mu_{1\to 1} + \psi^2) \\ &= 2\ln(1 + \psi^1 + \psi^2)\end{aligned} \tag{6.24}$$

where we substituted Z_{tj} with (6.17) and the messages in (6.24) with (6.21) and (6.22). From (6.23) and (6.24) we therefore get that the Bethe pseudodual is

$$\begin{aligned}F^{\#}_{B,SH} &= (m-1)\sum_{t=1}^{n} \ln Z_t + (n-1)\sum_{j=1}^{m} \ln Z_j - \sum_{t=1}^{n}\sum_{j=1}^{m} \ln Z_{tj} \\ &= \ln(1 + \psi^1 + \psi^2) - 2\ln(1 + \psi^1 + \psi^2) \\ &= -\ln(1 + \psi^1 + \psi^2).\end{aligned} \tag{6.25}$$

Finally, from our definition of the Bethe constant in (2.33), $Z_B = \exp(-F_B)$, if we use the Bethe pseudodual $F^{\#}_{B,SH}$ as the Bethe free energy F_B and negate and exponentiate (6.25), we see that the Bethe constant Z_B is

$$\begin{aligned} Z_B &= \exp(-F^{\#}_{B,SH}) \\ &= \exp(\ln(1 + \psi^1 + \psi^2)) \\ &= 1 + \psi^1 + \psi^2 \end{aligned}$$

which is equal to the expression we found for the exact normalization constant in (6.20), as expected.

6.2.1 Purpose for estimating the normalization constant

The present work delves deeper into the normalization constant than was done in [1] and the preceding project report for the following reasons. A central algorithm in current implementations is Murty's method which we discussed in Chapter 4.6.1. Perhaps the biggest draw-back to Murty's is that is only finds the *scores* of the posterior hypotheses, which are proportional to the probability of the posterior hypothesis. Therefore, after finding the N best hypotheses we have no guarantee that we have found a sufficiently large portion of the true probability distribution, although this is often the case. The worst case scenario is that the posterior hypotheses are uniformly distributed. In this case, approximating the posterior hypothesis distribution with the N best hypotheses will prune a prohibitively large portion of the posterior hypotheses, possibly with catastrophic consequences.

However, an accurate estimate of the normalization constant would provide us with a measure of how much of the probability mass we can represent with the posterior hypotheses enumerated by Murty's. This would allow us to adapt to cases where the posterior distribution is significantly "flat". It could also be used to terminate the algorithm early if we are confident that we can represent enough of the posterior distribution with the hypotheses enumerated thus far. In particular, for the last point, it is desirable to estimate an *underestimate* of the normalization constant, as we can then compare the sum over posterior hypotheses scores to the estimated constant and use the sum as our estimate of the true normalization constant if it is larger. If the estimate is larger than the true constant, then comparing the values gives little meaningful information about how well we represent the true distribution with the hypotheses found with Murty's.

7 | Efficient cluster marginalization

Multi-hypothesis trackers are required to prune the hypotheses that are kept to keep the computational load bounded [36]. In many cases, this removes negligible information from the system, as a relatively small proportion of the hypotheses holds most of the probability mass. The main basis for the use of Murty's method in multi-hypothesis trackers was precisely that enumerating the M most likely hypotheses suffices to capture the entire hypothesis distribution.

However, in case we have cluster merging, the resulting prior hypothesis space in general becomes so large that the assumption that the M best enumerated hypotheses from the posterior hypothesis space suffices becomes questionable.

Thus, if such a hypothesis enumeration scheme was instead used on the unmerged clusters, this would in principle cause considerably less information loss.

We can achieve this by exploiting the insight that in practice, in terms of probabilistic graphs, the dependency structure between the clusters is *sparse*. We can therefore avoid enumerating the merged hypothesis space by using a clever marginalization scheme, which is the topic of the following chapter.

7.1 Delegating variables

The key insight to achieve a much more light-weight marginalization is to utilize the fact that the prior clusters are *independent conditioned on the shared, gated measurements*. Such a shared, gated measurement will in the following be referred to as a *linking measurement*. We start by further overparameterizing the association factor graph by defining the *delegating variable* d_k^l for each linking measurement b_k^l with $l \in \mathcal{L}_k$ where $\mathcal{L}_k \subseteq \{1, \dots, m_k\}$ is the linking measurement index set containing the global measurement index of each linking measurement linking together clusters, all in timestep k. Define also the *per-cluster linking measurement index set* $\mathcal{L}_k^c$ containing the global

measurement indices that links cluster c to other clusters where it follows that

$$\mathcal{L}_k = \bigcup_{c=1}^{C} \mathcal{L}_k^c. \tag{7.1}$$

The purpose of introducing the delegating variable d_k^l is, as its name suggests, to delegate the measurement to exactly one cluster, or none of them. Additionally, define for each cluster c with a linking measurement the dummy measurement b_k^{cl} as the cluster local measurement that is delegated to the cluster. In practice, the variable d_k^l ensures that exactly one such local measurement b_k^{cl} exists, or none. As an example, consider the graphs in Figure 7.1 based upon Figure 5.2. We can identify measurement b^2 as the linking measurement, as it is gated by both track 3 in cluster 1 and track 4 and 5 in cluster 2. By now introducing d^2 as a variable delegating b^2 to either cluster 1 or 2 or none of them, we explicitly split the clusters. This way, when conditioning on d^2, the two clusters become independent, and so we avoid reenumerating the prior hypothesis space.

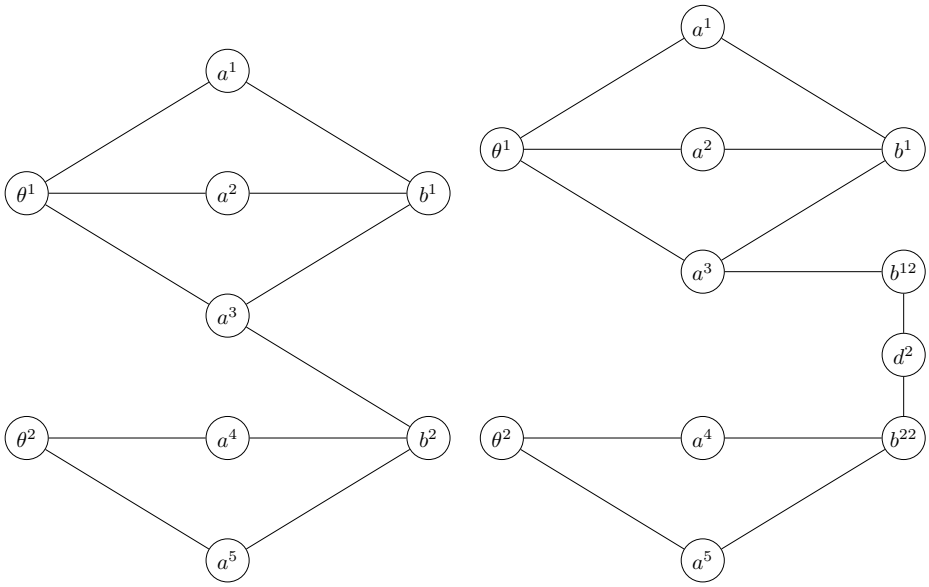


Figure 7.1: *Simplified multi-cluster association graph with and without the delegating variable d^2 for linking measurement b^2. The left figure shows the original association problem, while the right graph shows the overparameterized representation.*

7.2 The conditional marginalization procedure

Having set the stage in Chapter 7.1 for how to avoid reenumerating the prior hypotheses, we derive here the used equations. We will follow a divide-and-conquer approach, where we first break down the expression and then explain how to combine the pieces again to the full multi-cluster, multi-hypothesis marginals and normalization constant. A simplified diagram showing the full computation flow can be found in Figure 7.2 for reference.

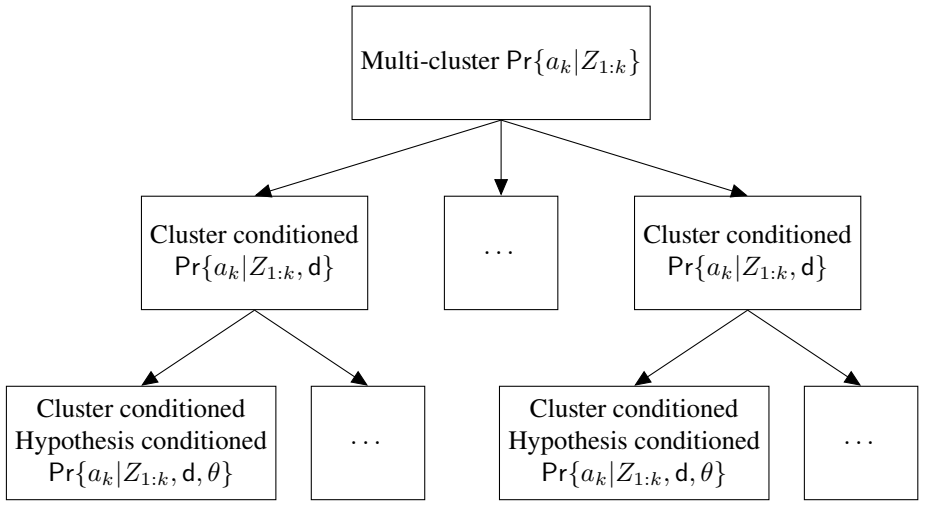


Figure 7.2: *Simplified illustration of how the cluster conditioning method can be visualized as a tree. We first start with the full multi-cluster posterior in the root node. By conditioning on the linking measurements, we restore independence between clusters which let us delegate the computation one level down to each cluster. Each cluster can then either compute the desired marginals and normalization constant at this level or further delegate the computation another level down by conditioning on the prior hypotheses. After each leaf node is finished, the results are propagated upwards the tree and combined in the root.*

For a given track t existing in prior cluster c, where we will allow the notation $t \in c$, we seek the posterior marginal $\mathsf{Pr}\{a_k^t \mid Z_{1:k}\}$. First, we will properly define the event space of d_k^l. Consider the linking measurement assignment variable b_k^l that the corresponding d_k^l is related to with event space $b_k^l \in \mathcal{B}_k$ and where $\mathcal{B}_k$ is given as

$$\mathcal{B} = \{0\} \cup \bigcup_{c=1}^{C} \mathcal{T}_k^c \tag{7.2}$$

$$= \{0, 1, \ldots, n_k\}, \tag{7.3}$$

$\mathcal{T}_k^c$ is the *track-in-cluster index set* given by

$$\mathcal{T}_k^c = \{t \mid t \in c\}. \tag{7.4}$$

and $b_k^l = 0$ signifies a new track. The event space of d_k^l is then a disjoint partitioning of $\mathcal{B}_k$ such that

$$d_k^l = \{\{0\}\} \cup \bigcup_{c=1}^{C} \{\mathcal{T}_k^c\}. \tag{7.5}$$

More concretely, in the example in Figure 7.1, we would have

$$b^2 \in \{0, 1, 2, 3, 4, 5\}, \tag{7.6}$$

$$d^2 \in \{\{0\}, \{1, 2, 3\}, \{4, 5\}\}, \tag{7.7}$$

or, more compactly with gating,

$$b^2 \in \{0, 3, 4, 5\}, \tag{7.8}$$

$$d^2 \in \{\{0\}, \{3\}, \{4, 5\}\}. \tag{7.9}$$

In other words, the possible outcomes of d_k^l are that the measurement is a new track or that some cluster *has to assign it to an existing track in the cluster*. Finally, define the Cartesian product over linking measurement indices of cluster c in the set $\mathcal{L}_k^c$ as $\mathsf{D}_k^c = \prod_{l \in \mathcal{L}_k^c} d_k^l$ such that an element $\mathsf{d} \in \mathsf{D}_k^c$ is a *tuple* of elements from each $d_k^l \in \mathcal{D}_k^c$ and unique. By total probability and Bayes' rule, the desired marginal can then be rewritten as

$$\mathsf{Pr}\{a_k^t \mid Z_{1:k}\} = \sum_{\mathsf{d} \in \mathsf{D}_k^c} \mathsf{Pr}\{a_k^t \mid Z_{1:k}, \mathsf{d}\} \, \mathsf{Pr}\{\mathsf{d} \mid Z_{1:k}\} \tag{7.10}$$

$$\propto \sum_{\mathsf{d} \in \mathsf{D}_k^c} \mathsf{Pr}\{a_k^t \mid Z_{1:k}, \mathsf{d}\} \, p(Z_k \mid \mathsf{d}, Z_{1:k-1}) \mathsf{Pr}\{\mathsf{d} \mid Z_{1:k-1}\} \tag{7.11}$$

$$\propto \sum_{\mathsf{d} \in \mathsf{D}_k^c} \mathsf{Pr}\{a_k^t \mid Z_{1:k}, \mathsf{d}\} \, p(Z_k \mid \mathsf{d}, Z_{1:k-1}) \tag{7.12}$$

where we use in (7.12) that a priori $\mathsf{Pr}\{\mathsf{d} \mid Z_{1:k-1}\} \propto 1$ is uniform, and can be moved into the proportionality sign. We argue that $\mathsf{Pr}\{\mathsf{d} \mid Z_{1:k-1}\}$ is uniform as, conditioned only on the previous measurement sets $Z_{1:k-1}$, we have no information on the measurement set Z_k. Thus, we also have no information about how the linking measurements $b_k^l \in \mathcal{L}_k$ should be delegated, and therefore all outcomes of d are equally likely.

7.2.1 Direct single-cluster, multihypothesis marginalization

At this point we have two options for how to proceed. The first option is that we have access to a single-cluster, multi-hypothesis solver which then can be applied directly to (7.12). We compute the conditioned marginals for each cluster independently due to independence to form $\mathsf{Pr}\{a_k^t \mid Z_{1:k}, \mathsf{d}\}$. The conditioned multi-cluster normalization constant $p(Z_k \mid \mathsf{d}, Z_{1:k-1})$ is given as simply the product over the conditioned single-cluster normalization constants,

$$p(Z_k \mid \mathsf{d}, Z_{1:k-1}) = \prod_{c=1}^{C} p(Z_k^c \mid \mathsf{d}, Z_{1:k-1}) \tag{7.13}$$

again due to conditional independence between the clusters, where we use the notation $p(Z_k^c \mid \mathsf{d}, Z_{1:k-1})$ to indicate the normalization constant of cluster c for in total C clusters.

7.2.2 Marginalization by total probability over hypotheses

It is, however, more common to have access to a single-hypothesis, single-cluster solver, e.g. a solver used in a JPDA filter. The second option is therefore that we further condition the marginals to sum over all prior hypotheses θ, which, again by total probability and Bayes' rule, yields

$$\mathsf{Pr}\big\{a_k^t \mid Z_{1:k}, \mathsf{d}\big\} = \sum_{\theta} \mathsf{Pr}\big\{a_k^t \mid Z_{1:k}, \mathsf{d}, \theta\big\}\, \mathsf{Pr}\{\theta \mid Z_{1:k}, \mathsf{d}\} \tag{7.14}$$

$$\propto \sum_{\theta} \mathsf{Pr}\big\{a_k^t \mid Z_{1:k}, \mathsf{d}, \theta\big\}\, p(Z_k \mid Z_{1:k-1}, \theta, \mathsf{d}) \mathsf{Pr}\{\theta \mid Z_{1:k-1}, \mathsf{d}\} \tag{7.15}$$

$$= \sum_{\theta} \mathsf{Pr}\big\{a_k^t \mid Z_{1:k}, \mathsf{d}, \theta\big\}\, p(Z_k \mid Z_{1:k-1}, \theta, \mathsf{d}) \mathsf{Pr}\{\theta \mid Z_{1:k-1}\} \tag{7.16}$$

where we have used that θ is independent of d a priori such that $\mathsf{Pr}\{\theta \mid Z_{1:k-1}, \mathsf{d}\} = \mathsf{Pr}\{\theta \mid Z_{1:k-1}\}$. This follows from the fact that d only contains information about which clusters should be assigned the linking measurements. Specifically, internal to a cluster, *there is no information about which track gets the measurements*, which implies there is no information in d about which hypothesis θ that is correct.

The marginal $\mathsf{Pr}\{a_k^t \mid Z_{1:k}, \mathsf{d}, \theta\}$ with associated normalization constant $p(Z_k \mid Z_{1:k-1}, \theta, \mathsf{d})$ can be computed with a single-cluster, single-hypothesis solver. From (7.16) we recog-

nize the conditioned single-cluster normalization constant $p(Z_k^c|\mathsf{d}, Z_{1:k-1})$ as

$$p(Z_k^c|\mathsf{d}, Z_{1:k-1}) = \sum_{\theta^c} p(Z_k^c \mid Z_{1:k-1}, \theta^c, \mathsf{d}) \mathsf{Pr}\{\theta^c \mid Z_{1:k-1}\}\,. \tag{7.17}$$

7.3 Improving performance with dynamic programming

The astute reader will notice that the sum in (7.12) is over $|\mathsf{D}_k^c|$ terms, which in general could make the repeated computation of $\mathsf{Pr}\{a_k^t \mid Z_{1:k}, \mathsf{d}\}$ infeasible for practical purposes. However, this is not the case. Since the clusters are independent given d_k^l, for some cluster c, knowing exactly which other cluster c' got a linking measurement b_k^l is irrelevant, only whether c got it or not. Hence, we can significantly improve the performance of the method with *dynamic programming*.

The cluster c can perform a binary look-up of which measurements are delegated and not delegated to it and do the computation only if that particular configuration has not been seen previously. The cluster can then cache the result and return it immediately if the marginal for an equal configuration is requested at a later point. This also holds for the estimated normalization constant. In terms of Figure 7.2, we can think of the naive, repeated computation as visiting all the leaf nodes of the computation tree. By using result caching we effectively prune most of the leaf nodes, reducing the number of computations.

As an example, consider the association scenario in Figure 7.3 where three single-track, single-hypothesis clusters are merging because of a single measurement b^1.

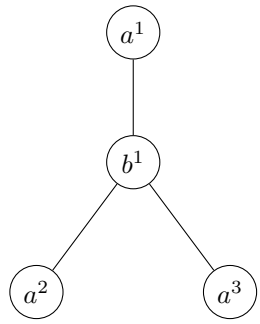


Figure 7.3: *Example scenario where caching results improve performance.*

In this case, the delegating variable d^1 has event space

$$d^1 = \{\{0\}, \{1\}, \{2\}, \{3\}\} \tag{7.18}$$

and the augmented association graph can be visualized as in Figure 7.4.

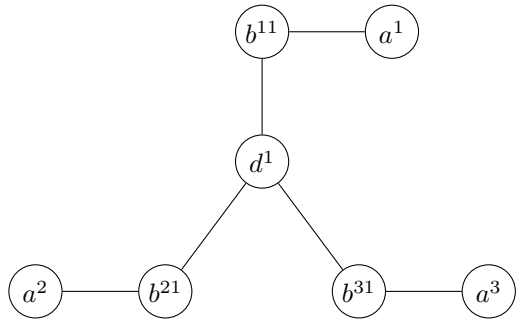
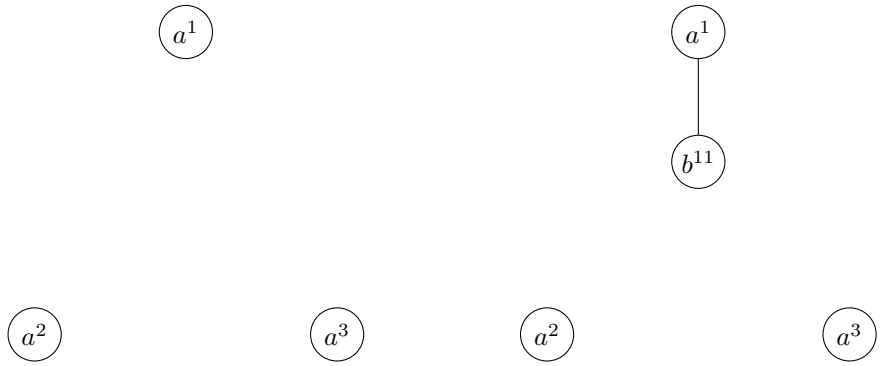


Figure 7.4: *Augmented association graph for result caching example.*

Let us focus on the cluster with track a^1, and consider the two first terms in the marginalization sum in (7.12). This corresponds to conditioning on $d^1 = \{0\}$ and $d^1 = \{1\}$, which results in the two conditional association graphs in Figure 7.5.

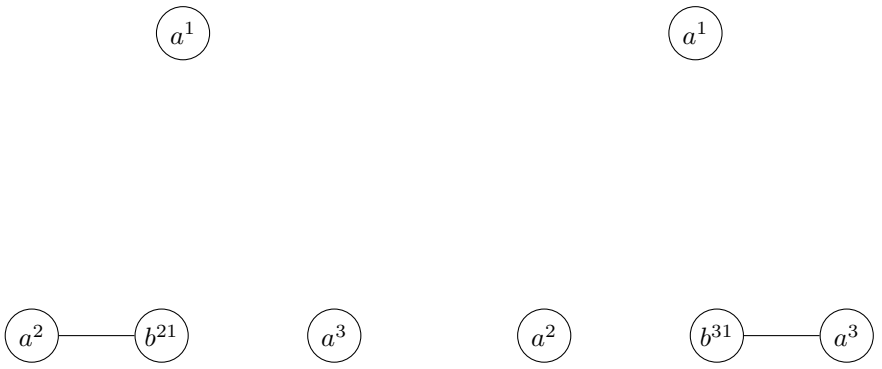


(a) *The first term, where $d^1 = \{0\}$.* ***(b)*** *The second term, where $d^1 = \{1\}$.*

Figure 7.5: *The conditional association graphs in the two first terms of the marginalization.*

In this case, the association cases relative a^1 are clearly different, as in the first case we have no detections while in the second we do. Consider now the two next terms, where $d^1 = \{2\}$ and $d^1 = \{3\}$, which are visualized in Figure 7.6.

Relative a^1, we see that the association cases in Figure 7.6 are identical to the first case it solved, i.e. the association case where $d^1 = \{0\}$. This follows from the fact that the clusters are *independent* when we condition on d^1, and so the association



(a) *The third term, where* $d^1 = \{2\}$. ***(b)*** *The fourth term, where* $d^1 = \{3\}$.

Figure 7.6: *The conditional association graphs in the second two terms of the marginalization.*

for a^1 is independent of a^2 and a^3. This then implies that the computed marginals and normalization constant must also be equal to the case where $d^1 = \{0\}$, and so the number of required computations is reduced from the naive 4 to just 2, halving the number of computations. In general, the number of required computations is $2^{|\mathcal{L}_k^c|}$, where we recall that $\mathcal{L}_k^c$ is the set of linking measurement indices for cluster c in timestep k. In the example above $\mathcal{L}^1 = \{1\}$ so cluster 1 is required to perform $2^1 = 2$ computations, as we previously concluded with.

7.4 Computing the exact solution by problem transposition

We will now concern ourselves with how we can adapt the above method to compute the exact solution with traditional single-cluster, single-hypothesis hypothesis enumeration. In typical implementations that compute the exact association marginals, we treat the association problem as figuring out what measurement to associate to the track association variables a^t. We can, of course, equivalently consider the association problem as *associating measurements to tracks*, i.e. figuring out what track to associate to measurement association variable b^j, which we will call *transposing the association problem*. This will be useful to do when we want to enforce that a measurement is associated to a track. Transposing the problem can be done the following way. Recall the joint association

posterior in (5.11), repeated here for convenience

$$\begin{aligned}\mathsf{Pr}\{\theta^r_{1:k}|Z_{1:k}\} &\propto \mathsf{Pr}\big\{\theta^l_{1:k-1}|Z_{1:k-1}\big\} \prod_{t\in M^l_k}\Big(1-r^{lt}_{k|k-1}P_D\Big) \prod_{t\in D^l_k}\frac{r^{lt}_{k|k-1}P_D l^{lt}}{\lambda+P_D\tilde{\nu}^{lt}_k} \\ &= \mathsf{Pr}\big\{a^1,\dots,a^n,\theta_{1:k-1}\mid Z_{1:k}\big\} \end{aligned} \tag{7.19}$$

that we derived using the track weights from the PMBM filter, and where the form in (7.19) is to emphasize the dependence on a^t. Normally we do marginalization on (7.19) to retrieve each $\mathsf{Pr}\{a^t \mid Z_{1:k}\}$. Let us now derive an expression for the joint distribution $\mathsf{Pr}\big\{b^1,\dots,b^m,\theta_{1:k-1}\mid Z_{1:k}\big\}$, for m measurements, by rewriting (7.19) into

$$\mathsf{Pr}\{\theta^r_{1:k}|Z_{1:k}\} \propto \mathsf{Pr}\big\{\theta^l_{1:k-1}|Z_{1:k-1}\big\} \prod_{t\in M^l_k}\Big(1-r^{lt}_{k|k-1}P_D\Big) \prod_{t\in D^l_k}\frac{r^{lt}_{k|k-1}P_D l^{lt}}{\lambda+P_D\tilde{\nu}^{lt}_k} \tag{7.20}$$

$$= \mathsf{Pr}\big\{\theta^l_{1:k-1}|Z_{1:k-1}\big\} \prod_{t\in M^l_k}\Big(1-r^{lt}_{k|k-1}P_D\Big) \frac{\prod_{t\in D^l_k}\Big(1-r^{lt}_{k|k-1}P_D\Big)}{\prod_{t\in D^l_k}\Big(1-r^{lt}_{k|k-1}P_D\Big)}$$

$$\cdot \prod_{t\in D^l_k}\frac{r^{lt}_{k|k-1}P_D l^{lt}}{\lambda+P_D\tilde{\nu}^{lt}_k} \tag{7.21}$$

$$= \mathsf{Pr}\big\{\theta^l_{1:k-1}|Z_{1:k-1}\big\} \prod_{t\in M^l_k\cup D^l_k}\Big(1-r^{lt}_{k|k-1}P_D\Big)$$

$$\cdot \prod_{t\in D^l_k}\frac{r^{lt}_{k|k-1}P_D l^{lt}}{\big(\lambda+P_D\tilde{\nu}^{lt}_k\big)\Big(1-r^{lt}_{k|k-1}P_D\Big)} \tag{7.22}$$

$$= K(\theta^l_{1:k-1})\mathsf{Pr}\big\{\theta^l_{1:k-1}|Z_{1:k-1}\big\} \prod_{t\in D^l_k}\frac{r^{lt}_{k|k-1}P_D l^{lt}}{\big(\lambda+P_D\tilde{\nu}^{lt}_k\big)\Big(1-r^{lt}_{k|k-1}P_D\Big)} \tag{7.23}$$

where we introduced the hypothesis-dependent constant

$$K(\theta^l_{1:k-1}) = \prod_{t\in M^l_k\cup D^l_k}\Big(1-r^{lt}_{k|k-1}P_D\Big). \tag{7.24}$$

This constant disappears into the proportionality sign when computing hypothesis-conditioned marginals. Only when we compute the measurement-oriented, hypothesis-conditioned normalization constant Z_b which we want to convert back to the track-oriented, hypothesis-conditioned normalization constant Z_a do we need $K(\theta^l_{1:k-1})$ be-

cause of the relation

$$Z_a = K(\theta^l_{1:k-1})Z_b. \tag{7.25}$$

With the measurement-oriented assocation posterior in (7.23) we define the unary factors $\tilde{\psi}^j(t)$ for b^j in the modified factor graph as

$$\tilde{\psi}^j(b^j = t) = \begin{cases} \frac{r^{lt}_{k|k-1} P_D l^{lt}}{\left(\lambda + P_D \tilde{\nu}^{lt}_k\right)\left(1 - r^{lt}_{k|k-1} P_D\right)}, & t = 1, \dots, n \\ 1, & t = 0 \end{cases}, \tag{7.26}$$

where $b^j = 0$ denotes that the measurement is a false alarm. When we enforce that some measurement j has to be associated to a track we simply use $\tilde{\psi}^j(0) = 0$ to assign 0 probability to the event that the measurement is a false alarm, which implies that it is a detection of an existing track with probability 1.

Doing this transposing of the hypothesis-conditioned association problem will in turn give us the marginals $\mathsf{Pr}\{b^j \mid Z_{1:k}, \theta, \mathsf{d}\}$. These can then easily be converted back to the desired track marginals $\mathsf{Pr}\{a^t \mid Z_{1:k}, \theta, \mathsf{d}\}$ by using

$$\mathsf{Pr}\{a^t = j \mid Z_{1:k}, \theta, \mathsf{d}\} = \mathsf{Pr}\{b^j = t \mid Z_{1:k}, \theta, \mathsf{d}\}, \quad j = 1, \dots, m \tag{7.27}$$

$$\mathsf{Pr}\{a^t = 0 \mid Z_{1:k}, \theta, \mathsf{d}\} = 1 - \sum_{j=1}^{m} \mathsf{Pr}\{a^t = t \mid Z_{1:k}, \theta, \mathsf{d}\}. \tag{7.28}$$

7.5 Alternative event space definition

Transposing the problem in order to disallow false alarms will work when using an exact solver, as hypothesis enumeration becomes the same regardless of whether the association problem is solved with respect to a^t og b^j. However, a critical fault when using LBP to approximate the marginals is that when we *enforce* the association of a measurement to a track we in practice turn the SNR for a measurment infinitely large. As previously mentioned, in [8] they experienced inaccurate estimates with LBP when the SNR was excessively large, which we also experienced. In addition, the convergence guarantees they make do not hold any longer. Thus, we propose the following approximation that gives more stable LBP estimates.

Instead of using the disjoint event space defined in (7.5) we choose the overlapping partitioning of the event space

$$d^l_k = \{\{0\}\} \cup \bigcup_{c=1}^{C} \{\{0\} \cup \mathcal{T}^c_k\}. \tag{7.29}$$

In this case, the example in (7.9) becomes instead

$$b^2 \in \{0, 3, 4, 5\}, \tag{7.30}$$

$$d^2 \in \{\{0\}, \{0, 3\}, \{0, 4, 5\}\}. \tag{7.31}$$

Although it is still possible to perform the exact summation over the entire event space with this partitioning, we now need to compensate for overlap, meaning we have to do summation by the *inclusion-exclusion principle*. This follows from the fact that each cluster now contributes to the misdetection event. Consider the sum in (7.12). When we have overlap between the events in the tuple d_k^l, we need to, by the inclusion-exclusion principle, perform increasingly larger *intersections* of the events and alternate between adding and subtracting the computed marginals and likelihoods. To put it more concretely, we will here sketch out how the computation can be done for a simple test case. See Figure 7.7 for illustration of the association case.

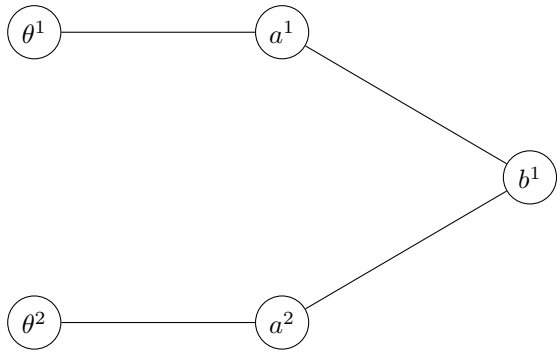


Figure 7.7: *Test case for showcasing the inclusion-exclusion principle.*

In this case, we get that the event space of d^1 is given as

$$d^1 = \{\{0\}, \{0, 1\}, \{0, 2\}\}. \tag{7.32}$$

Since we are summing over the entire event space of d^1, which is nothing more than the union of the different outcomes, the sum in (7.12) will be denoted for simplicity as $\mathsf{Pr}\{\{0\} \cup \{0, 1\} \cup \{0, 2\}\}$ before partitioning. For the first "stage" of the summation,

we do the same sum as in (7.12), i.e. we sum over the events

$$\{0\}$$
$$\{0,1\}$$
$$\{0,2\}.$$

In the next step we subtract the overlap, which is done by subtracting the intersections of the events. We have $\binom{3}{2} = 3$ different combinations, which are the events

$$\{0\} \cap \{0,1\} = \{0\}$$
$$\{0\} \cap \{0,2\} = \{0\}$$
$$\{0,1\} \cap \{0,2\} = \{0\}.$$

We have one final stage to do before the entire event space is exhausted, which is adding again the intersection of all the events. This corresponds to the $\binom{3}{3} = 1$ way

$$\{0\} \cap \{0,1\} \cap \{0,2\} = \{0\}.$$

In conclusion, the sum becomes

$$\begin{aligned}\mathsf{Pr}\{\{0\} \cup \{0,1\} \cup \{0,2\}\} &= \mathsf{Pr}\{\{0\}\} + \mathsf{Pr}\{\{0,1\}\} + \mathsf{Pr}\{\{0,2\}\} \\ &\quad - \mathsf{Pr}\{\{0\} \cap \{0,1\}\} - \mathsf{Pr}\{\{0\} \cap \{0,2\}\} - \mathsf{Pr}\{\{0,1\} \cap \{0,2\}\} \\ &\quad + \mathsf{Pr}\{\{0\} \cap \{0,1\} \cap \{0,2\}\} \\ &= \mathsf{Pr}\{\{0\}\} + \mathsf{Pr}\{\{0,1\}\} + \mathsf{Pr}\{\{0,2\}\} - 2\mathsf{Pr}\{\{0\}\}.\end{aligned}$$

The above procedure was tangible as we could manually enumerate all the intersections of each inclusion-exclusion stage. In the general case, however, such enumeration becomes exponentially complex to do. We note that due to the structure of the problem, when we intersect two or more events, either the events have to be identical to become the same, e.g. $\{0,1\} \cap \{0,1\} = \{0,1\}$ or they become $\{0\}$. This structure arises from the fact that tracks are exclusive to a single cluster, so intersecting events across clusters must reduce to the false alarm event. Utilizing this insight should allow for more efficient enumeration. Coming up with an expression or algorithm for doing this enumeration efficiently by utilizing the structure of the problem is left as future work. Instead, in this work we approximate the sum by only using the first stage, such that in the example

above we approximate it as

$$\mathsf{Pr}\{\{0\} \cup \{0,1\} \cup \{0,2\}\} \approx \mathsf{Pr}\{\{0\}\} + \mathsf{Pr}\{\{0,1\}\} + \mathsf{Pr}\{\{0,2\}\}$$

and then run LBP internally for each conditional outcome, for example $\{0\}$, $\{0,1\}$ and $\{0,2\}$.

7.5.1 Approximation errors from Bonferroni inequalities

The approximation errors from removing "stages" from the inclusion-exclusion sum can be described by the *Bonferroni inequalities* [48]. In particular, the inequalities state that, for an odd number of stages, we always *overestimate*, while for an even number we always *underestimate*. In the example above, this means that we are guaranteed that

$$\mathsf{Pr}\{\{0\} \cup \{0,1\} \cup \{0,2\}\} \leq \mathsf{Pr}\{\{0\}\} + \mathsf{Pr}\{\{0,1\}\} + \mathsf{Pr}\{\{0,2\}\}$$

since we only add one stage, i.e. an odd number of stages. How this affects the estimated marginals is unclear since we always normalize them in the end anyway. However, this means that we will consistently *overestimate the true Bethe constant* when using this approximation. Intuitively, we interpret this error as keeping the overlap between the events, thus overcounting by adding extra probability mass to our unnormalized marginals and normalization constant.

As a last note, the Bonferroni inequalities say nothing about how large the error is and does not relate the error from using an odd number of stages to the error when using an even number. It does, however, guarantee us a decreasing error as we add more stages of the same parity. As an example, using five stages accumulates less error than three which accumulates less than using one. However, we are not guaranteed that using six stages accumulates less error than using five.

7.6 Three novel variations using LBP

We now present three novel variations of how to use LBP with cluster conditioning. The first method is to use single-cluster, multi-hypothesis LBP directly to compute $\mathsf{Pr}\{a_k^t \mid Z_{1:k}, \mathsf{d}\}$ with the message definitions in Lemma 1 on the corresponding factor graph. Given the messages at the fixed point we approximate the cluster-conditioned normalization constant $p(Z_k^c | Z_{1:k-1}, \mathsf{d})$ by using the Bethe pseudodual given in Theorem 2.

The two other variations are based on hypothesis conditioning, where we perform LBP on the single-cluster, single-hypothesis association graph in the same way as was

done in [8] to estimate the association marginals. In this case we are required to compute the hypothesis-conditioned normalization constant $p(Z_k^c|Z_{1:k-1}, \theta^c, \mathsf{d})$. The first way we propose is the same as was done in the preceding project report and [1], where we use a Poisson approximation for the true, Binomial likelihood $p(Z_k^c|Z_{1:k-1}, \theta^c)$ similarly to what is done in the PHD filter. The exact derivation details are outside the scope of this text, and the result is that we can use the approximation

$$p(Z_k|\theta, Z_{1:k-1}) \approx K \exp\left(-\sum_{t=1}^{n_k} r_k^t P_D\right) \prod_{j=1}^{m_k} \left[\left(\sum_{t=1}^{n_k} \frac{r_k^t P_D l^{jt}}{\lambda + P_D \tilde{\nu}_k^{jt}}\right) + 1\right] \tag{7.33}$$

where we recognize $r_k^t P_D l^{jt}/(\lambda + P_D \tilde{\nu}_k^{jt})$ as the detection weight from the association posterior in (5.11), $r_k^t P_D = 1 - (1 - r_k^t P_D)$ as related to the misdetection weight also in (5.11), and K is some constant that is the same across all hypotheses θ and is therefore eventually cancelled out. The other approach is estimating the hypothesis-conditioned normalization constant with the Bethe pseudodual in Corollary 2.

III

Results

8 | Method evaluation on simple test case

Before delving into the results of running the proposed methods on a large dataset, the following chapter will test the MCMH-LBP method on selected, simple test cases in order to better assess the dynamics of LBP on the factor graphs that we encounter in practice. Although multiple methods for inference have been presented so far, for brevity and also focus on the dynamics of LBP, the following will concern itself with only MCMH-LBP as the approximate method.

8.1 Definition of data structures used

Before proceeding with our discussion, and later present results, we first need to define two central data structures used in the implementations for data association.

8.1.1 The reward matrix

The first is the *reward matrix* $\mathbf{R}$. We will use the convention that the reward matrix is a real matrix with shape $n_k \times (m_k + 1)$, i.e. that $\mathbf{R} \in \mathbb{R}^{n_k \times (m_k+1)}$. Each row corresponds to a track a^t and each element on that row corresponds to either the *logarithm* of detection likelihood when associating the track with a measurement b^j or the logarithm of the misdetection probability. We use the first column to stack the log misdetection probabilities and the remaining columns for detection loglikelihoods. Therefore, the element R_{uv} at row u and column v is defined as

$$R_{uv} = \begin{cases} \ln(1 - r^{jt}_{k|k-1} P_D), & v = 1 \\ \ln(r^{jt}_{k|k-1} P_D l^{jt}) - \ln(\lambda + P_D \tilde{\nu}^{jt}_k), & v = 2, \ldots, m_k + 1 \end{cases} \tag{8.1}$$

where $t = u$ and $j = v - 1$ for $u = 1, \ldots, n_k$ and $v = 1, \ldots, m_k + 1$.

8.1.2 Prior hypotheses distribution

The second data structure is how the prior hypotheses are structured. Each hypothesis denotes the tracks existing conditioned on it, which can be none, together with the probability. As an example, for a simple case of two tracks a^1 and a^2 existing in each their hypothesis together with an empty hypothesis, all with uniform probability, we will use a table as

$$\begin{array}{l|ll} \theta^1 = 1 & [1] & 1/3 \\ \theta^1 = 2 & [2] & 1/3 \\ \theta^1 = 3 & [\,] & 1/3 \end{array} .$$

Lastly, marginals are given as tables also, where each row is the marginal distribution for a track a^t, the first probability is misdetection, the last is nonexistence and the middle are detection. An example for two tracks and one measurement might be

	0	1	N
a^1	0.339	0.661	0.000
a^2	0.280	0.321	0.399

8.2 Analysis strategy and options

Since many results have already been established for a single-hypothesis association graph in previous literature [8], [26], the following will test different perturbations of the *hypothesis parameters* to test how the estimates from MCMH-LBP react to different configurations.

We have some options available in this regard. Choosing a uniform prior hypothesis distribution gives the association problem a stronger multi-hypothesis flavor. The opposite would be if a single hypothesis has probability 1, in which case the problem reduces to single-hypothesis.

Additionally, each track can in the multi-hypothesis formulation take the nonexistence value $a^t = N$. The marginal probability for this event is related to how the tracks are distributed in the different prior hypotheses. As an example, if a track exists in all prior hypotheses then we can immediately conclude that $\mathsf{Pr}\{a^t = N \mid Z_{1:k}\} = 0$, i.e. the

track has to exist. Testing how estimates are affected by varying the track distribution across prior hypotheses will therefore be interesting to do.

Lastly, evidence that favors a particular prior hypothesis should depend on if a track in that prior hypothesis is detected. If a track has a large likelihood for being detected, then we expect the probability of the prior hypotheses where it exists to increase. Testing this can be done by changing the values of the reward matrix. Changing the detection likelihoods would also introduce more loops into the association graph, which is interesting as well from a more general LBP perspective.

8.3 Testing and discussion on test case

We first inspect how LBP performs on the association graph given in Figure 8.1, same as the example in Figure 5.2.

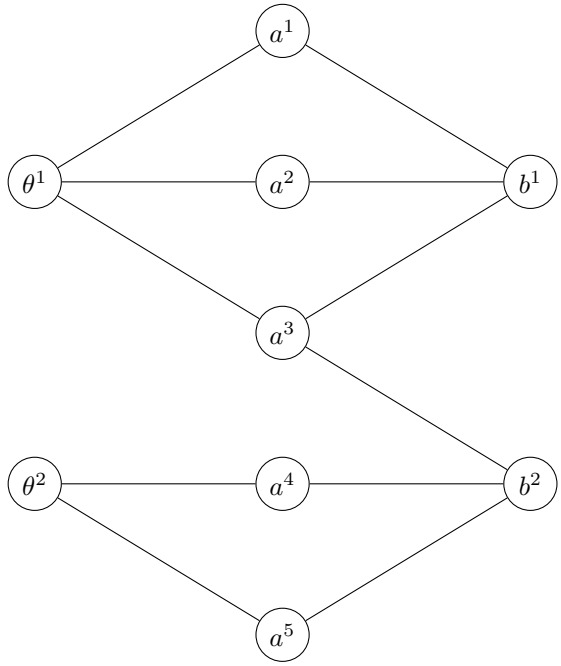


Figure 8.1: *Simplified association graph of test case with two clusters, five tracks and two measurements.*

We arbitrarily choose the following parameters to do inference on. The reward matrix

is given as

$$\mathbf{R} = \begin{bmatrix} -0.6 & 3.0 & -\infty \\ -0.56 & 3.2 & -\infty \\ -0.46 & -3.0 & 1.2 \\ -0.62 & -\infty & 3.0 \\ -0.55 & -\infty & -0.4 \end{bmatrix} \tag{8.2}$$

and the prior hypotheses are

$\theta^1 = 1$	$[1, 2]$	0.5
$\theta^1 = 2$	$[1, 3]$	0.5
$\theta^2 = 1$	$[4]$	0.5
$\theta^2 = 2$	$[5]$	0.5

.

With these parameters, the exact marginals are given as

	0	1	2	N
a^1	0.341	0.659	0.000	0.000
a^2	0.282	0.322	0.000	0.396
a^3	0.312	0.001	0.084	0.604
a^4	0.063	0.000	0.842	0.096
a^5	0.067	0.000	0.028	0.904

with exact normalization constant $Z = 228.528$. In this case, the LBP marginals are

	0	1	2	N
a^1	0.339	0.661	0.000	0.000
a^2	0.281	0.321	0.000	0.399
a^3	0.310	0.000	0.088	0.601
a^4	0.066	0.000	0.859	0.075
a^5	0.071	0.000	0.004	0.925

with Bethe constant $Z_B = 222.945$, in other words underestimating the true normalization constant. We see that in this case, LBP is a more than satisfactory estimate of the true distribution, with a strong correlation between estimated and exact marginal probabilities.

Interestingly, if now modify the hypothesis distribution of cluster 1 such that we instead have

$$\begin{array}{l|cc} \theta^1 = 1 & [1,2,3] & 0.5 \\ \theta^1 = 2 & [\,] & 0.5 \\ \hline \theta^2 = 1 & [4] & 0.5 \\ \theta^2 = 2 & [5] & 0.5 \end{array}.$$

the exact marginals become

	0	1	2	N
a^1	0.520	0.433	0.000	0.047
a^2	0.445	0.508	0.000	0.047
a^3	0.751	0.001	0.201	0.047
a^4	0.117	0.000	0.734	0.150
a^5	0.125	0.000	0.024	0.850

with normalization constant $Z = 116.075$. The LBP marginals, however, become

	0	1	2	N
a^1	0.246	0.468	0.000	0.286
a^2	0.211	0.503	0.000	0.286
a^3	0.556	0.001	0.157	0.286
a^4	0.097	0.000	0.793	0.110
a^5	0.105	0.000	0.006	0.890

with Bethe constant $Z_B = 177.565$, i.e. in this case significantly *overestimating* the normalization constant. We are able to identify that indeed the prior cluster, cluster 1, where we have an empty hypothesis is the culprit, as LBP is able to estimate the marginals for tracks in cluster 2 well. When we test the opposite case, i.e.

$$\begin{array}{l|cc} \theta^1 = 1 & [1,2] & 0.5 \\ \theta^1 = 2 & [1,3] & 0.5 \\ \hline \theta^2 = 1 & [4,5] & 0.5 \\ \theta^2 = 2 & [\,] & 0.5 \end{array}.$$

we get the exact marginals

	0	1	2	N
a^1	0.318	0.682	0.000	0.000
a^2	0.261	0.299	0.000	0.440
a^3	0.289	0.001	0.150	0.560
a^4	0.079	0.000	0.743	0.179
a^5	0.798	0.000	0.023	0.179

and constant 149.413. With LBP we get marginals

	0	1	2	N
a^1	0.323	0.677	0.000	0.000
a^2	0.266	0.304	0.000	0.430
a^3	0.294	0.000	0.136	0.570
a^4	0.071	0.000	0.720	0.209
a^5	0.722	0.000	0.068	0.209

and constant 157.234. It is interesting to see that we in this case also overestimate the normalization constant, but not by much. Also, the marginals are much better behaved. This observation suggests that it is not empty hypotheses in itself that makes LBP ill-behaved. Our first suspicion is that this is related to the fact that track a^3 links together the clusters. This is because the we can either think of the last perturbation as generating an empty hypothesis or moving the particular track a^3 into a different prior hypothesis. If generating an empty hypothesis is not the reason then this suggests it is instead related to track a^3, which happens to be the track that links the two prior clusters. We therefore change the reward matrix to

$$\mathbf{R} = \begin{bmatrix} -0.600 & 3.000 & -\infty \\ -0.560 & 3.200 & -\infty \\ -0.460 & -3.000 & -\infty \\ -0.620 & -\infty & 3.000 \\ -0.550 & 1.000 & -0.400 \end{bmatrix}$$

which corresponds to the graph in Figure 8.2.

Doing the same tests again reveals that we get the same behavior as before, i.e. that cluster 2 is more or less unaffected by how the tracks are distributed over hypotheses, but that cluster 1 is heavily affected by it. We speculate therefore that instead the estimation

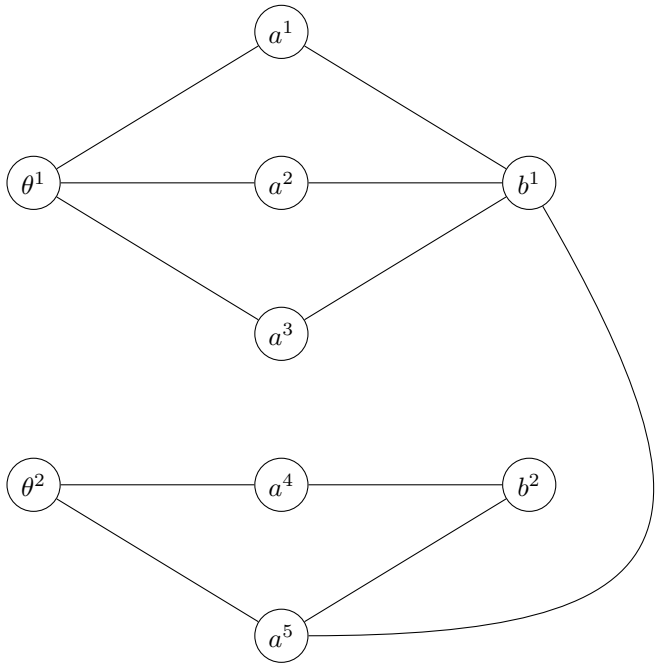


Figure 8.2: *Simplified association graph of test case with two clusters, five tracks and two measurements. This alternative has changed where the link across the two prior clusters occur.*

accuracy is related to the fact that in cluster 1 we have three tracks competing for a measurement, inducing more loops in the graph. For cluster 2 there is only one loop internally,

$$\theta^2 \to a^5 \to b^2 \to a^4 \to \theta^2,$$

while we strictly speaking have three loops for cluster 1,

$$\theta^1 \to a^3 \to b^1 \to a^2 \to \theta^2,$$
$$\theta^2 \to a^3 \to b^1 \to a^1 \to \theta^2,$$
$$\theta^2 \to a^2 \to b^1 \to a^1 \to \theta^2.$$

At this point, we make the following postulate based on our observations so far. When assuming only the tracks $[1, 2]$ or $[1, 3]$ exist, we only have two competing tracks, which reduces the number of loops. This then improves approximated marginals. To test our postulate, we return to our original association case, but use the prior hypotheses

distribution

$$\begin{array}{c|cc} \theta^1 = 1 & [1,2] & 0.5 \\ \theta^1 = 2 & [3] & 0.5 \\ \hline \theta^2 = 1 & [4] & 0.5 \\ \theta^2 = 2 & [5] & 0.5 \end{array}.$$

The rational for doing this is that we for cluster 1 either have a single loop, in hypothesis $\theta^1 = 1$, or no loop, in hypothesis $\theta^1 = 2$. The results we get are the exact marginals

	0	1	2	N
a^1	0.528	0.440	0.000	0.033
a^2	0.452	0.516	0.000	0.033
a^3	0.024	0.002	0.006	0.967
a^4	0.028	0.000	0.912	0.060
a^5	0.030	0.000	0.030	0.940

and constant 142.710, with LBP giving

	0	1	2	N
a^1	0.289	0.467	0.000	0.244
a^2	0.247	0.509	0.000	0.244
a^3	0.190	0.000	0.054	0.756
a^4	0.051	0.000	0.892	0.058
a^5	0.054	0.000	0.003	0.942

and constant 195.921. Evidently, removing loops by placing tracks in specific prior hypotheses does not seem to help, as the results are similar to the case where all tracks were placed in the same prior hypothesis.

Lastly, we inspect tweaking the reward matrix. Specifically, we have so far only inspected a multi-cluster scenario with a single link. If we repeat the previous example,

but with the reward matrix

$$\mathbf{R} = \begin{bmatrix} -0.600 & 3.000 & 3.000 \\ -0.560 & 3.200 & 3.200 \\ -0.460 & -3.000 & 1.200 \\ -0.620 & 3.000 & 3.000 \\ -0.550 & -0.400 & -0.400 \end{bmatrix},$$

then all tracks compete for all measurements. In other words, the association graph between tracks and measurement is maximally dense, inducing a large number of loops. Curiously, this results in the marginals

	0	1	2	N
a^1	0.261	0.347	0.347	0.044
a^2	0.224	0.366	0.366	0.044
a^3	0.012	0.000	0.032	0.956
a^4	0.243	0.255	0.226	0.276
a^5	0.260	0.008	0.008	0.724

and constant 575.868 for the exact case, and marginals

	0	1	2	N
a^1	0.225	0.333	0.328	0.114
a^2	0.196	0.348	0.342	0.114
a^3	0.100	0.000	0.014	0.886
a^4	0.202	0.285	0.282	0.231
a^5	0.217	0.007	0.007	0.769

and constant 556.944 for LBP, meaning that, arguably, forming a more dense graph actually *improved the overall accuracy* in this case. We note in particular that track a^3 seems to have the weakest estimates and that the Bethe constant is much more accurate than previously, now just barely underestimating the true constant rather than overestimating it. This is in particular strange considering LBP should in general be more accurate for sparser graphs, as the variables are less correlated with each other. This warrants further investigation.

8.4 Summary of observations

In this example we inspected a simple two-cluster example with three tracks in one cluster and two in the other and two measurements. We discovered that for the initial configuration of prior hypotheses and reward matrix parameters, LBP and the Bethe constant were suitable approximations to the exact marginals and normalization constant, respectively. However, tweaking just slightly the distribution of tracks over the prior hypotheses in cluster 1 drastically changed the estimated marginals from LBP due to overestimating the nonexistence probabilities. The Bethe constant also overestimated the true normalization constant. It was speculated that this had to do with having an empty hypothesis, but this seemed to not be the case when testing with an empty hypothesis in cluster 2. It was then believe that it could be related to the track 3 in cluster 1 that caused the link between the clusters. This, however, did also not seem to be the case as changing the link to a different track made no significant difference. Lastly, it was speculated that it could be related to the number of competing tracks and more importantly, the number of loops in the cluster graph. Distributing the tracks to remove loops conditioned on the prior hypothesis did not seem to help either. In the last test we inspect how increasing the number of links between prior clusters affect the estimation accuracy. Spectacularly, for the case it was tested on, this improved accuracy, warranting further testing.

In conclusion, it is hard to draw any consistent conclusion from the presented observations. It is clear that mainly cluster 1 is the root of the estimation problems. Since cluster 2 seems more robust to estimation errors despite trying to provoke the same faults as for cluster 1, it could be simply related to the difference in number of tracks in the two clusters.

9 | Introduction to the dataset used for testing

The methods proposed to approximate marginals, as discussed in Chapters 6 and 7, were evaluated on a large simulated dataset. This dataset, named "9 ravens," consists of 1397 radar scans in 2 dimensions. In this scenario, there are 8 actual targets, while the radar is installed on a separate vehicle. To test data association, the association cases were generated using several *Monte Carlo* simulations on the same radar scans, resulting in a total of 9188 association cases. More detailed information about this dataset can be found in [49].

The following chapter intends to introduce statistics about the dataset used for testing in order to better understand the data association scenarios that the proposed methods are expected to handle.

9.1 Overview of track clusters statistics

An interesting metric for data association, and in particular when using LBP, is the *number of tracks competing for the same measurement*. This is first and foremost because it is when tracks compete for measurements that we experience the true combinatorial complexity of the problem. In addition, from a graphical point of view, we close loops in the association graph when multiple tracks compete for the same measurement, and so we expect LBP to perform worse the more measurement contention we have. In Figure 9.1 we present a histogram over the number of competing tracks in the same hypothesis. We choose to measure contention this way as two tracks that gate the same measurement, but does not exist in the same hypothesis, are in some sense not aware of each other, and so they only indirectly compete for the measurement. From Figure 9.1 we see that the number is of competing tracks is moderately low, and so we expect LBP to not be too

affected by it.

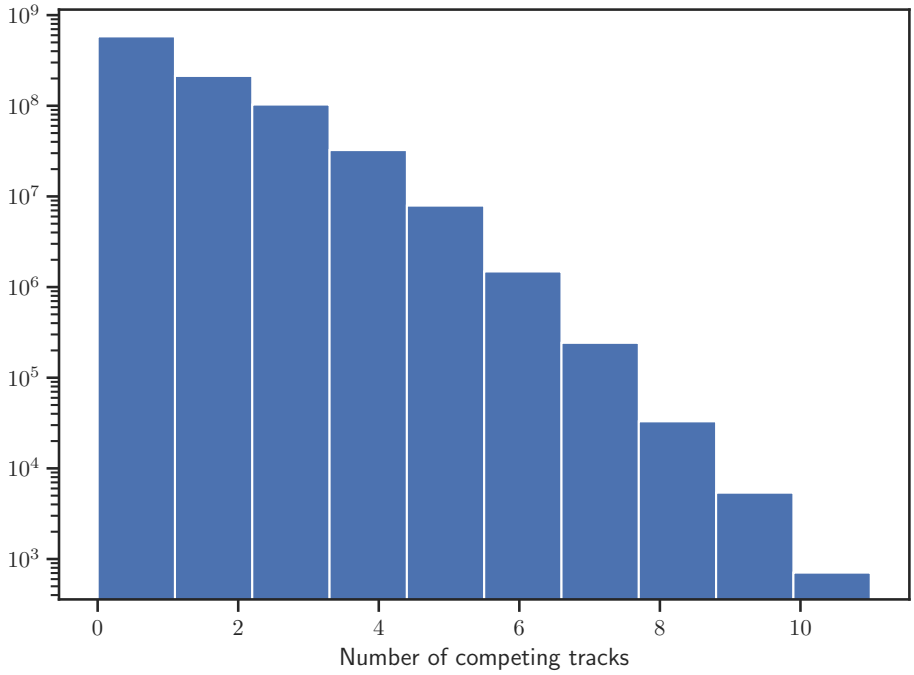


Figure 9.1: *Histogram over number of tracks in the same hypothesis competing for a measurement. Note the logscale on the y-axis.*

Since most of the proposed methods are based on more efficiently to compute marginals when clusters merge, we will in this section inspect some relevant statistics for the cluster merging scenarios. A summary of selected metrics can be found in Table 9.1.

Metric	Quantity
Total number of posterior clusters	81439
Total number of superclusters	14623
Average number of linking measurements	2.663
Average number of prior clusters in supercluster	3.081
Average number of prior hypotheses in superclusters	5566.104
Average number of prior hypotheses in prior cluster in supercluster	23.395

Table 9.1: *Table summarizing selected metrics from the cluster merging cases in the dataset.*

In Figure 9.2 we provide a scatter plot between the number of prior hypotheses after cluster merging and the corresponding average number of tracks in the prior hypotheses. The main take away from this plot is that there is a considerable number of cases where

the number of prior hypotheses explodes, further emphasizing the point that enumerating the posterior hypotheses from so many prior hypotheses is unfeasibly costly.

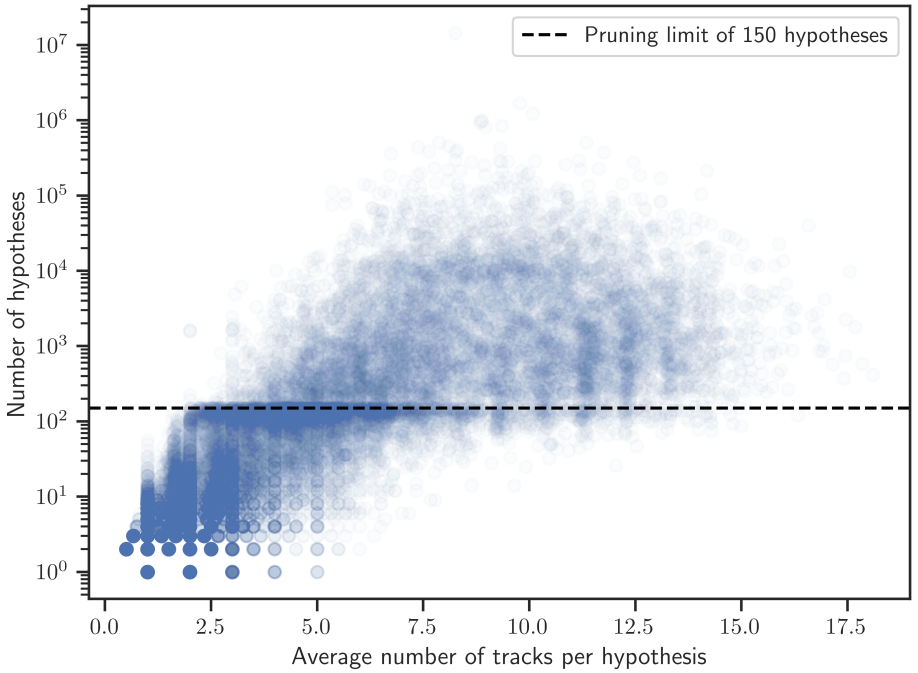


Figure 9.2: *Scatter plot displaying the distribution over number of hypotheses in a cluster and the corresponding average number of tracks in each hypothesis. The pruning limit is an implementation parameter that determines how many hypotheses that are kept to the next timestep. In other words, all points above this line must necessarily be from superclusters where the reenumerated number of prior hypotheses is greater than this threshold. Note the logscale on the y-scale.*

Since the computational complexity in the cluster-conditioning methods is bounded by the number of linking measurements for each prior cluster in the supercluster, we have in Figure 9.3 provided a histogram over the number of linking measurements that a cluster in a merging scenario has. The main take away here is that, at least for this dataset, the number of computations is more than feasible with an average of 2.663.

The histogram in Figure 9.4 shows the distribution of numbers of prior clusters in a supercluster. From this figure it is clear that we in practice do not expect too many clusters to combine into a supercluster each timestep, which also significantly bounds the computational complexity of doing cluster-conditioned marginalization.

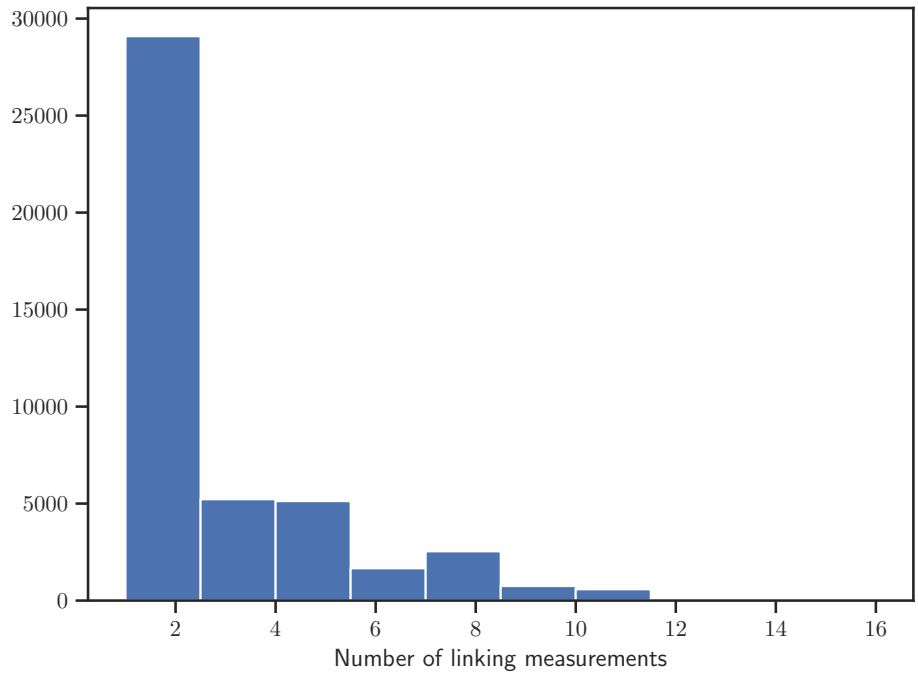


Figure 9.3: *Histogram over number of linking measurements that linked a cluster to others in a merging scenario.*

9.2 The methods compared

We compare four different methods in the following results. The first, main method is the multi-cluster, multi-hypothesis LBP method directly on the full association graph called MCMH-LBP, presented in Chapter 6. Based on the cluster-conditioning method presented in Chapter 7, and in particular the approximated version described in Chapter 7.5, we then also test three variations. The first is called "Efficient MHLBP" and uses MH-LBP to do inference on the conditioned, single-cluster, multi-hypothesis association posterior. The cluster normalization constants are approximated using the Bethe pseudodual in Theorem 2. The two last methods are "Approx Efficient PHD" and "Approx Efficient Bethe" which both do further hypothesis conditioning of the marginals and approximates them using the same LBP approach as described in [8]. The hypothesis-conditioned likelihood in "Approx Efficient PHD" uses the PHD approximation while in "Approx Efficient Bethe" the Bethe pseudodual in Corollary 2 is used.

The main two methods of interest are MCMH-LBP and "Approx Efficient Bethe". The method MCMH-LBP is interesting as it is one of the main results of this thesis, but

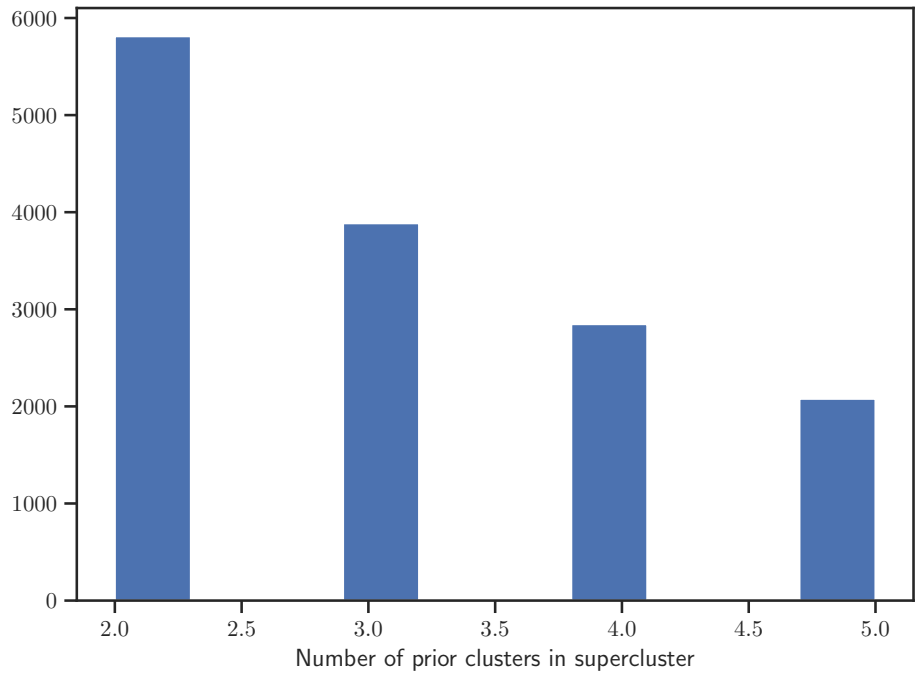


Figure 9.4: *Histogram over number of prior clusters in the superclusters in the dataset.*

also because it would be promising to use in a real tracker because of its computational efficiency and relatively easy implementation. We are also interested in understanding better LBP dynamics, especially when used in a multi-hypothesis setting. The other method, "Approx Efficient Bethe", is interesting in particular from a more theoretical point of view. Since the cluster and hypothesis conditioning are exact operations, disregarding the inclusion-exclusion approximation, the only approximate part of the algorithm is the single-cluster, single-hypothesis LBP. Because of the analysis and proofs from [8] and [26] with regard to e.g. convergence of LBP on the single-cluster, single-hypothesis association graph, this method therefore offers more guarantees and is expected to be more robust in a practical implementation. Additionally, contrary to the PHD variation, we expect the Bethe constant to be a better approximation of the true normalization constant as it is computed from a function we expect to be closer to the true distribution than the Poisson distribution.

Due to the novelties of the presented work, no true benchmark exists that the methods can be compared to. We therefore choose as a bona-fide benchmark to use Murty's method that we introduced in Chapter 4.6.1. Let the set $\mathscr{M}_k^c$ denote the posterior hypotheses

enumerated by Murty's in some cluster c and timestep k. Let l denote the hypothesis index for some posterior hypothesis $\theta^l_{1:k}$. We allow the notations $l \in \mathscr{M}^c_k$ and $\theta^l_{1:k} \in \mathscr{M}_k$ to both denote the same as the indices are unique. Let also $w^l_k \propto \mathsf{Pr}\{\theta^l_{1:k} \mid Z_{1:k}\}$ denote the hypothesis weight. The Murty constant Z^c_M for cluster c is then calculated as

$$Z^c_M = \sum_{l \in \mathscr{M}^c_k} w^l_k \tag{9.1}$$

and the multi-cluster constant Z_M

$$Z_M = \prod_{c=1}^{C} Z^c_M. \tag{9.2}$$

Each Murty association marginal $\tilde{p}_M(a^t)$ for track t in cluster c is calculated as

$$\tilde{p}_M(a^t = j) = \sum_{l \in \mathscr{M}^c_k \,:\, a^{lt} = j} \frac{w^l_k}{Z^c_M} \tag{9.3}$$

where $\sum_{l \in \mathscr{M}^c_k \,:\, a^{lt}=j}$ denotes the sum over all hypotheses where the association $a^t = j$ is made for $j \in \{0, 1, \ldots, m_k\}$ with m_k measurements. We then set the nonexistence probability to

$$\tilde{p}_M(a^t = N) = 1 - \sum_{j=0}^{m_k} \tilde{p}_M(a^t = j). \tag{9.4}$$

Lastly, an important parameter in Murty's is the maximum number of posterior hypotheses to enumerate. We denote this number by K and call the method for different parameter choices "K-Murty". If no K is indicated we use $K = 150$ for computational reasons.

10 | 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 9.

10.1 Normalization constant estimation accuracy

A comparison of the different ways to estimate the normalization constant is found in Figure 10.1.

We see that all the methods are able to somewhat estimate the normalization constant well, with MCMH-LBP performing the best. Not only do we observe that it has the lowest variance out of all the methods, but that it consistently underestimates the normalization constant, which the preferable behavior as discussed in Chapter 6.2.1. In fact, in our results MCMH-LBP only overestimates the true normalization constant in 21 cases out of the 9138. It is interesting to see that largest errors in the estimated normalization constant from MCMH-LBP are the cases where it overestimates it. This must be related to similar observations made in Chapter 8.3.

Murty's method, here with 150 posterior hypotheses enumerated, performs the best. This is probably due to the sheer number of cases where we properly capture the true hypothesis distribution with just 150 posterior hypotheses.

The error induced by the approximation discussed in Chapter 7.5.1 can be seen in all the cluster-conditioning based method estimates, in particular for "Approx Efficient Bethe". We know from the discussion in Chapters 2.3.2 and 2.3.4 that for the hypothesis-conditioned single-cluster association graph, the Bethe free energy upper-bounds the true free energy function. We therefore would expect, from $Z_B \leq Z$, i.e. (2.34), that the normalization constant estimates underestimate the true normalization constants, since the method is otherwise exact. As the estimates clearly overestimate instead, this must follow from the approximation we made according to the discussion in Chapter 7.5.1 regarding Bonferroni inequalities, i.e. that only adding a single stage of inclusion-

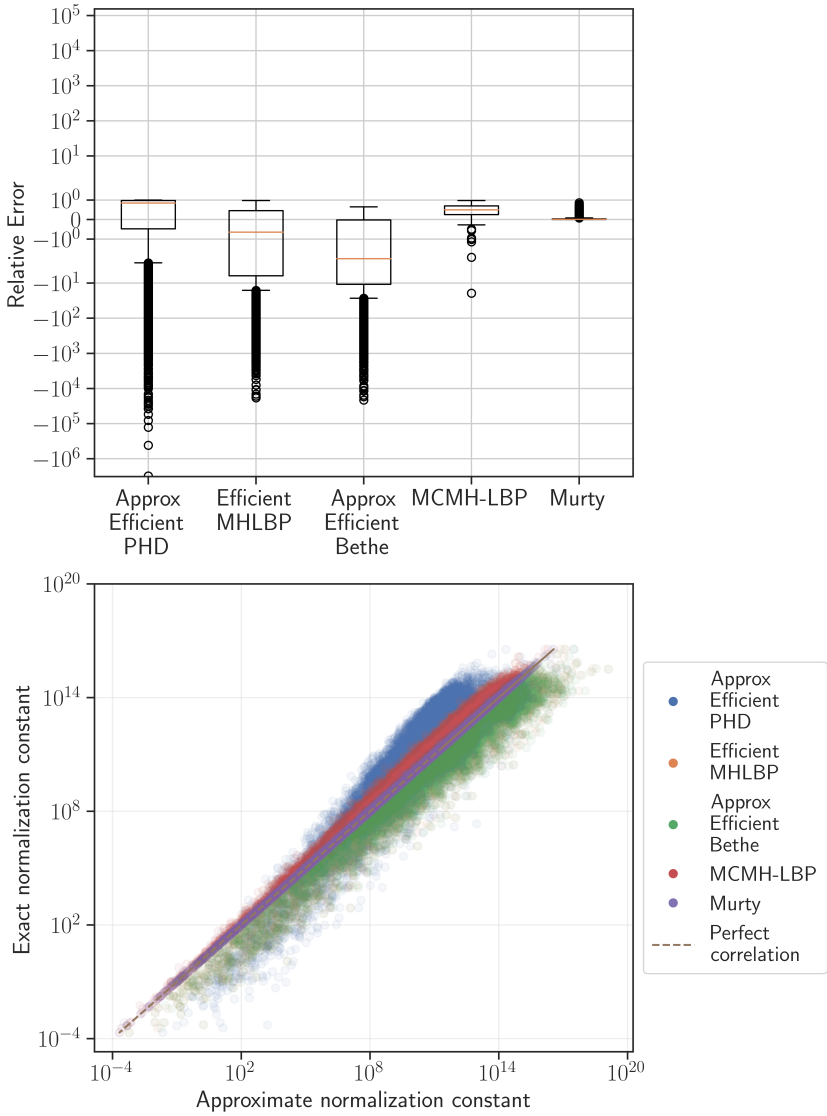


Figure 10.1: *Two plots visualizing the estimation accuracy of the estimated normalization constants. The top plot shows a boxplot of the relative error of the estimates, computed as, with Z as exact constant and $\tilde{Z}$ the approximate, $(Z - \tilde{Z})/Z$. The bottom plot shows a scatter plot of the different estimates relative the exact value.*

exclusion indeed gives us an overestimate. We do not have the same guarantees about the true estimated normalization constant when using the PHD constant or multi-hypothesis Bethe, but we see especially in the boxplot that they indeed are heavily overestimated.

The normalization constant estimated by "Approx Efficient PHD" seems to be correl-

ated to the true normalization constant, although we do observe a tendency to overestimate small constants and underestimate large constants. This is a natural consequence do to the Poisson approximation we make, as the Poisson distribution indeed is flatter than the true Binomial distribution.

10.2 Approximate marginals accuracy

Heatmap correlation plots of the approximated marginals for each of the tested methods can be found in Figure 10.2.

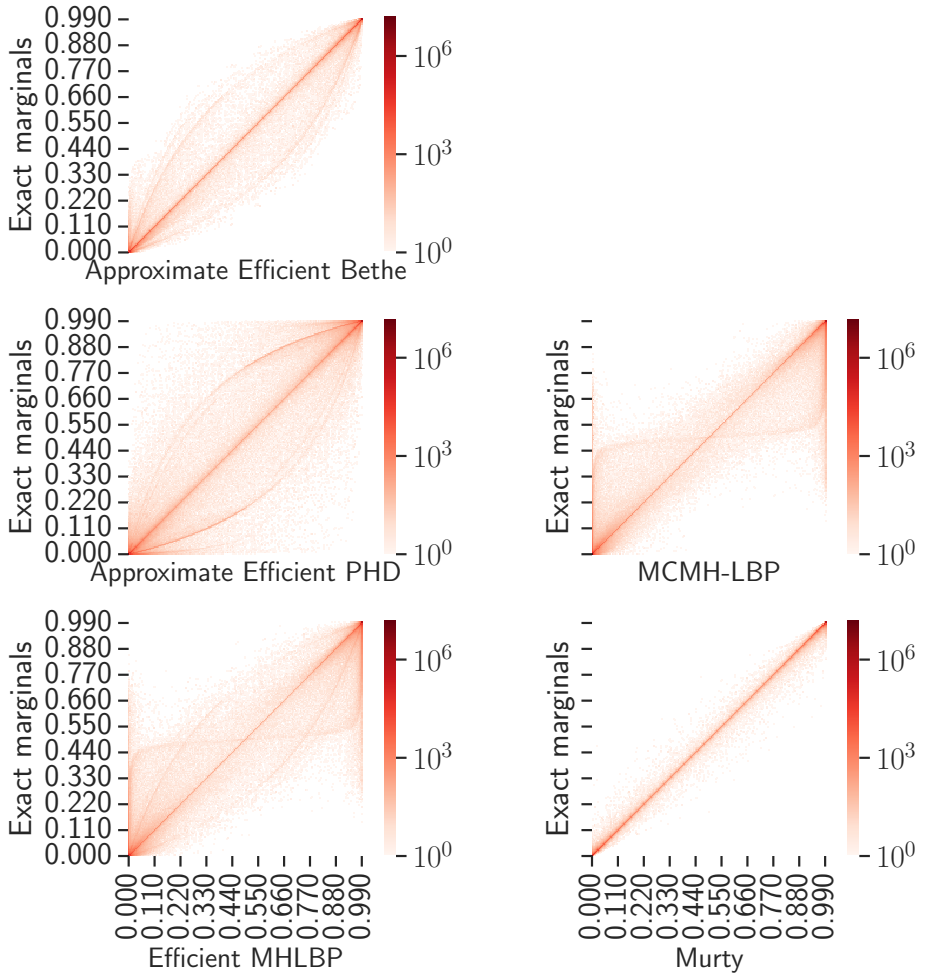


Figure 10.2: *Heatmap correlation plots of the different approximate marginalization methods.*

In particular two patterns in Figure 10.2 deserves attention. The first can be seen from comparing “Approximate Efficient Bethe”, “Approximate Efficient PHD” and “Efficient MH-LBP”, as they are all display the same variance ellipsis around the correlation line, which speculate is because they are all affected by approximation in the cluster-conditioning sum. Additionally, “Approximate Efficient PHD” features a slightly larger variance, which is probably due to use the PHD constant instead of Bethe.

The second pattern is that we can observe the effects of multi-hypothesis marginalization by comparing MCMH-LBP with “Efficient MH-LBP”. In these two plots the marginals are distributed in a transposed “S”-shape, where this shape is superimposed on top of the inclusion-exclusion variance for “Efficient MH-LBP”. Perhaps the most important consequence of this is that this means that multi-hypothesis LBP tends to make *radical* estimates, i.e. that the true marginals are pushed towards the extreme points 0 and 1. We can identify a line on both the left and right side of the correlation plots indicating that there is a significant portion of the estimates that are close to 0 or 1 when the true probability can be anywhere in between. As an example, we see the line on the right side of the MCMH-LBP plot stops at about 0.2 on the exact axis, indicating that exact probabilities as low as 0.2 were estimated to be close to 1. For the hypothesis-conditioned methods “Approximate Efficient Bethe” and PHD we do not see the same behavior, which strongly favors such approaches for more robust marginal estimation. In particular, the “Approximate Efficient Bethe” method seems to perform the best out of the presented methods.

The Murty’s method, here also with 150 as maximum number of posterior hypotheses enumerated, is again the overall best method. This is consistent with how well it estimates the normalization constant. Again, this is probably due to the sheer number of association cases where the distribution is so peaked that finding the 150 most likely hypotheses suffices to capture the entire distribution.

10.3 Survival function of marginal errors

The *survival function* to the empirical distribution of the different marginal errors can be found in Figure 10.3. We quickly describe how to interpret such a plot. For a given point at a curve, the y-value indicates the remaining proportion of the data that is larger than the corresponding x-value. Therefore, since these are errors, the most desirable is to have the curve hit the x-axis the earliest possible and also be below the other curves. The x-value at which the curve hits the x-axis is the largest error in the data, which we want to bound. Also, when the curve of a method is below the curve of another method,

this implies that the former method has less large errors than the latter method.

Overall, we see that the errors of the most of the methods seem to be similar, with the largest deviations for misdetection. In fact, all the presented methods except "Approximate Efficient Bethe" exhibit nearly identical behavior for all marginal errors except for misdetection. In particular, notice that the "MCMH-LBP" method, the only method that does not do cluster conditioning, catches up with "Approximate Efficient Bethe". There is a reasonable explanation for this. Recall from our overlapping definition of the event space of the delegating variable d that we duplicate the false alarm event for each cluster in addition to the null event where no cluster gets any linking measurement. The detection probabilities are much more well-behaved as they are only counted once for each track in each cluster. The misdetection probability, however, accumulates more error as each cluster accounts for the event that a linking measurement is a false alarm, which propagates into the misdetection probability.

By comparing "Max errors" to "Nonexistence errors", we see that the curves for the three methods in question are almost identical in the two plots. This suggests that the similar errors are related to the estimation of the nonexistence probability, as the large error seemingly distorts the distributions similarly for the three methods. The fact that "MCMH-LBP" and "Efficient MH-LBP" are so similar is reasonable as they both perform LBP on a multi-hypothesis association graph.

Considering the marginals are computed almost identically in "Approximate Efficient Bethe" and "Approximate Efficient PHD", it is interesting that the accuracy is so different. Considering the normalization of marginals is done after marginalization over the prior hypotheses for the two methods, it is definitively clear that the largest source of error is in that step. This shows that the Bethe constant indeed is a better approximation of the true hypothesis-conditioned likelihood than the PHD constant.

Lastly, it is important to note that "Approximate Efficient Bethe" performs significantly better than all the other methods, as we can tell from the fact that its curve hits the x-axis first. This fact further supports to use a hypothesis-conditioned method for marginal estimation. It still does not beat Murty, which still performs the best.

10.4 Inspection of the prior hypotheses posteriors

One significant observation made so far is how the marginal errors of "Approx Efficient Bethe" are clearly better than "MCMH-LBP" and "Efficient MH-LBP". This is interesting because they are all LBP and Bethe constant-based, but "Approx Efficient Bethe" does hypothesis-conditioned LBP, i.e. LBP does not marginalize over the prior hypothesis

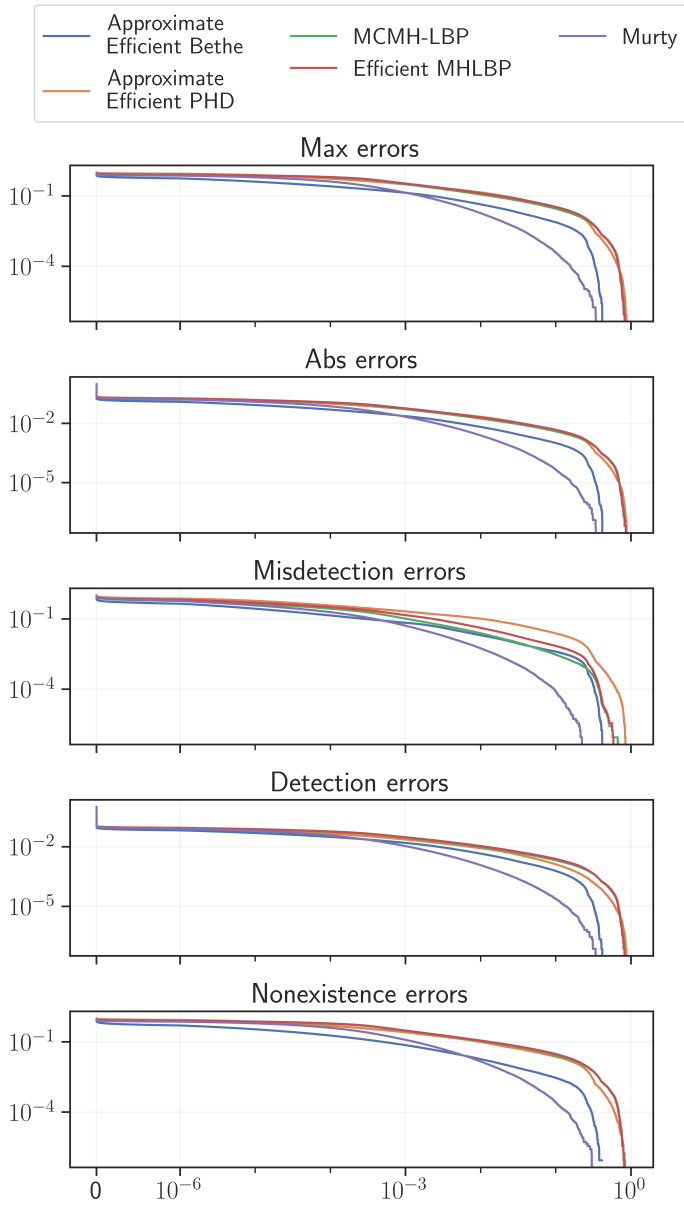


Figure 10.3: *Survival function for different marginal errors comparing the different methods tested. All plots show the absolute value of the marginal errors. The "Max errors" plot is over the maximum error for each estimated track marginal distribution. The "Abs errors" plot contains all errors. The "Misdetection errors" plot contains only the misdetection errors, and similarly for "Detection errors" and "Nonexistence errors". The symlog scaling is used on the x-axis, with a linear threshold of* 10^{-6}*. The y-axis is logscale.*

variable, while both "MCMH-LBP" and "Efficient MH-LBP" do. One observation in particular was the similarity between the "Max errors" and "Nonexistence errors" curves in Figure 10.3 that seemed to suggest that the estimation of the nonexistence probability is a large source of error.

Therefore, to investigate this discrepancy we will in the following section try to evaluate how the *prior hypothesis posterior* $\mathsf{Pr}\{\theta_{1:k-1} \mid Z_{1:k}\}$ is estimated by the different methods. This is because of the deep connection between the prior hypothesis posterior and the nonexistence marginals. As an example to build intuition, if a track is assigned a high nonexistence probability, then we have evidence that the track should not existence, and we are inclined to assign low probability to prior hypotheses that contain that particular track.

By inspecting the estimation accuracy of the prior hypothesis posterior we hope in particular to shed light on what failure modes LBP suffers from when used in a multi-hypothesis setting, as we have observed so far that hypothesis-conditioned methods seem to be more accurate and robust. See Figure 10.4 for correlation plot for the prior hypotheses posterior, $\mathsf{Pr}\{\theta_{1:k-1} \mid Z_{1:k}\}$, for the different methods tested.

The main observation we can make from Figure 10.4 is that they in large terms behave the same as the track association marginals. An important observation is that for the multi-hypothesis-based methods, they tend to *overestimate* the true probability, which we can tell from the fact that there is a larger density of points below the correlation line than above, particularly for exact probabilities above 0.5. We can see clear lines, especially on the right of the plot, for both "MCMH-LBP" and "Efficient MH-LBP". Thus, multi-hypothesis LBP favors overestimating the probabilities. The hypothesis-conditioned methods does not seem to exhibit the same behavior. Specifically, "Approximate Efficient Bethe" seems to be doing a particularly good job at estimating the prior hypothesis posterior with little variance. These observations support our suspicion that the prior hypothesis posterior indeed has a significant influence on the overall track association marginal estimation accuracy.

In order to better understand how the errors are distributed we provide a histogram and boxplot showing the signed error distribution for the prior hypotheses posterior estimates in Figure 10.5. Clearly, the two methods "Efficient MH-LBP" and MCMH-LBP have a bias towards overestimation, consistent with what we saw in the correlation plots. Again, we also see that the hypothesis-conditioned methods are unbiased in their estimates, where "Approximate Efficient Bethe" has the lowest variance.

It is clear from the above results that multi-hypothesis LBP suffers from overestimation of probabilities in the prior hypotheses posterior, and that this propagates into the

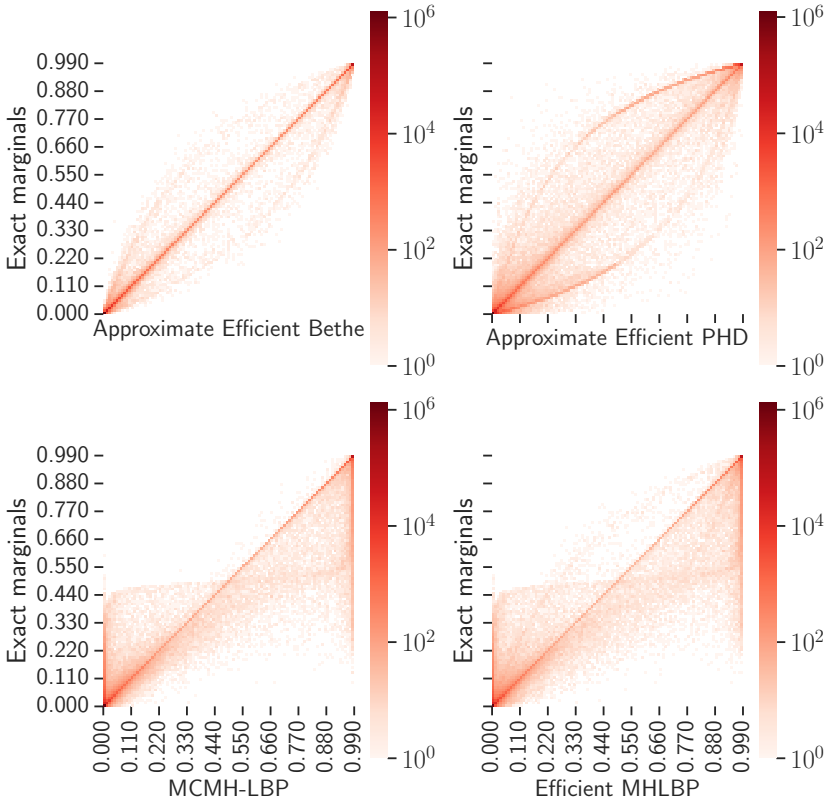


Figure 10.4: *Correlation plot for the prior hypotheses posterior* $\Pr\{\theta_{1:k-1} \mid Z_{1:k}\}$ *for the different methods tested.*

track association marginals. Since these probabilities are related to the nonexistence probability for the track association marginals, it would seem this is the largest source of error when estimating the association marginals with a multi-hypothesis-based LBP method.

We make one final remark on the matter with regard to MCMH-LBP. Despite the marginals being affected by the poor prior hypothesis posterior estimate, the estimated normalization constant seems not, considering MCMH-LBP by far is the most accurate at estimating the constant. This suggests that the inclusion-exclusion approximation we made is a larger source of error than first anticipated. This shows promise for "Approximate Efficient Bethe" to reliably estimate the normalization constant if the inclusion-exclusion sum can be performed exactly and efficiently.

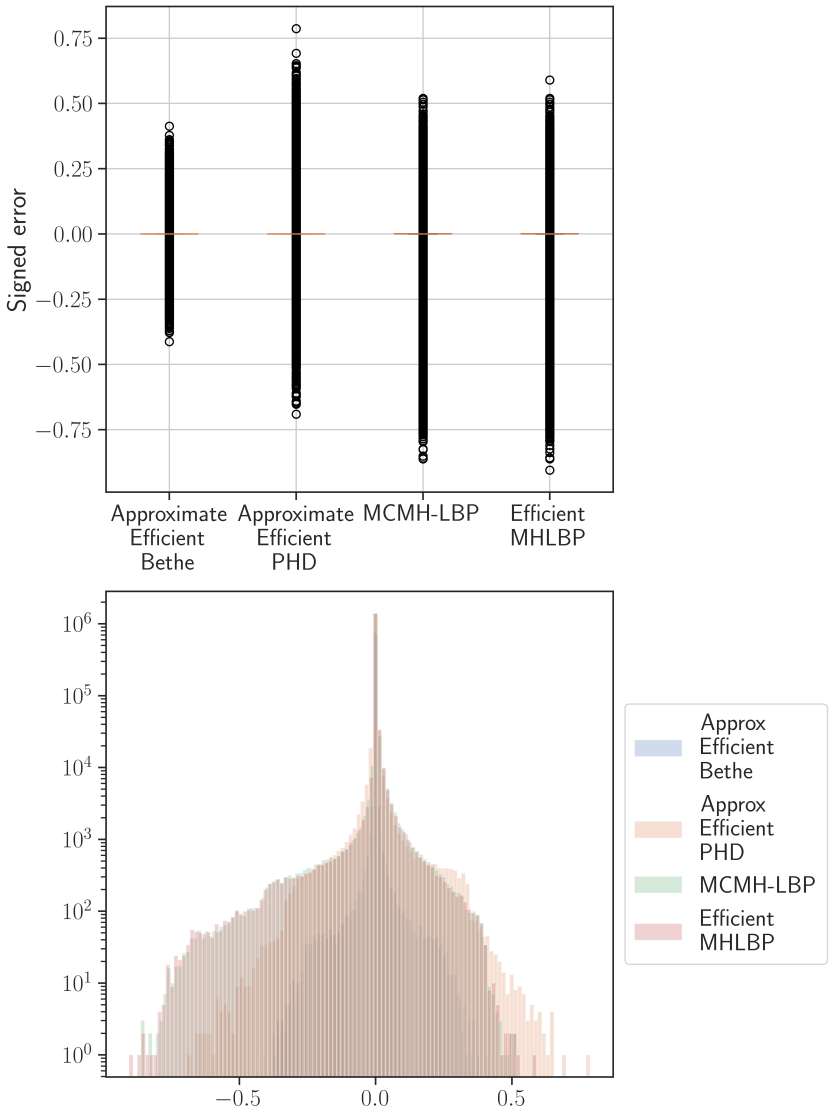


Figure 10.5: *Two plots visualizing the same data distributions over signed error for the estimates of the prior hypothesis posterior. With p denoting the exact posterior and $\tilde{p}$ an estimate, the error is computed as $p - \tilde{p}$. Note that the y-axis on the histogram is logscaled.*

10.5 Relating LBP to K-Murty for different K

Another important observation we have made so far is that Murty's method with $K = 150$ performs very well, retrieving an almost perfect solution in most situations. However, this

does not automatically mean we should discard the presented methods. Enumerating up to 150 is done to be more confident we actually propagate sufficient hypothesis probability mass to make sure we properly represent the multi-hypothesis probability distribution. Therefore, without recycling or any track cardinality balance, we are in practice required to use e.g. $K = 150$.

Enumerating fewer hypotheses, on the other hand, is preferable for several reasons. Firstly, it is cheaper for Murty's to enumerate fewer hypotheses. Secondly, keeping fewer hypotheses in the MBM component is cheaper to propagate under the prediction and update step. Thirdly, keeping more hypotheses in the MBM component means in general we have more tracks in the MBM component as well, which takes more memory resources. Therefore, it is relevant to investigate the benefits of using LBP for other values of K.

To investigate we have calculated marginals and normalization constants for $K \in \{10, 20, 50, 100, 150\}$. We compare the results to the normalization constant estimates to "MCMH-LBP" and the association marginals to "Efficient Approx Bethe", as these are the best results we have with LBP. In Figure 10.7 is a survival function plot over the different marginal errors for the Murty configurations. In Figure 10.6 there is a boxplot over the relative errors of the estimated normalization constants for the Murty methods and "MCMH-LBP".

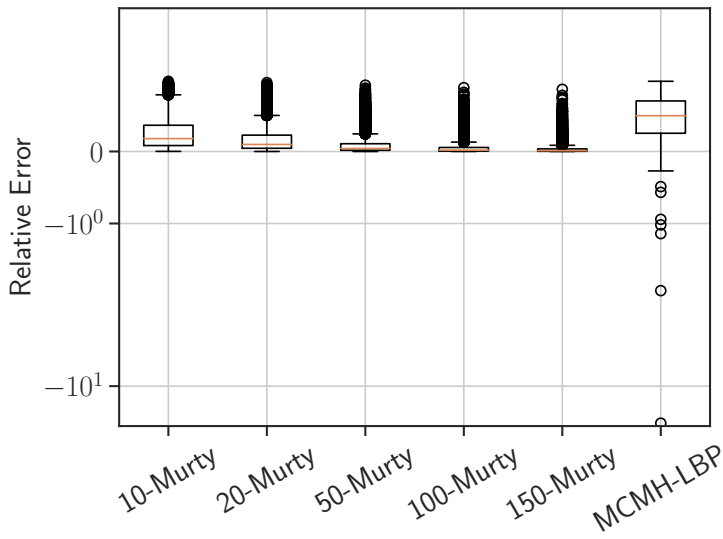


Figure 10.6: *Boxplot showing relative error of the estimated normalization constants. Relative error is computed as $(Z - \tilde{Z})/Z$ where Z and $\tilde{Z}$ are the exact and approximate normalization constants, respectively. The different configurations of Murty, i.e. the maximum number of hypotheses that Murty's is allowed to enumerate, is indicated by K in K-Murty.*

It is clear from Figure 10.6 that the posterior hypothesis distribution is often very peaked, which we can tell from the fact that even 10-Murty is able to very accurately approximate the normalization constant in most cases. We also see that all the K-Murty methods outperform "MCMH-LBP".

The main observation from Figure 10.7 is that Murty's is estimating the marginals well in a large number of cases, even for small K. The marginal errors from "Efficient Approx Bethe" seems to overall be bounded by the marginal errors of "10-Murty", except for misdetection. Overall, "Efficient Approx Bethe" is comparable to K-Murty, but in most cases, K-Murty performs the best. From these results, it would seem that more work is necessary to beat Murty's method.

10.5.1 Artificially peaked hypothesis distributions

Before we end this discussion, one important point regarding the performance of the Murty methods needs to be elaborated on. The fact that all of the Murty methods performed this well hinged on the fact that the posterior hypothesis distribution was sufficiently peaked, such that the K best posterior hypotheses held almost all the probability mass. The dataset that was used had already pruned the hypothesis space from the previous timestep with precisely Murty's method, and so no prior cluster contained more than 150 prior hypotheses. When unenumerated probability mass is pruned, the enumerated probability mass must be scaled up to sum to 1. This scaling makes existing peaks more peaked. Therefore, it is plausible that this scheme introduces *bias* towards some hypotheses, which then makes the posterior hypothesis distribution artificially peaked. If we had access to the true posterior hypothesis distribution, it might be more flat, such that the performance of Murty's method is more comparable to LBP.

As a simple example to show this bias, assume we have three true posterior hypotheses with weights 3, 2 and 1. The normalized probabilities are then $1/2$, $1/3$ and $1/6$. The difference between the most and second most probable hypotheses is then $1/6$. If we were to prune the last hypothesis, the pruned distribution instead becomes $3/5$ and $2/5$ where the difference is now $1/5$, i.e. larger than the actual difference $1/6$. In general, the difference between any two peaks in the original distribution is scaled by the ratio of the true normalization constant and the approximated, which necessarily always must be greater than 1, and equal iff we do no pruning. Thus, this example shows how accumulated pruning of hypotheses can in principle cause the hypothesis distribution to become artificially peaked.

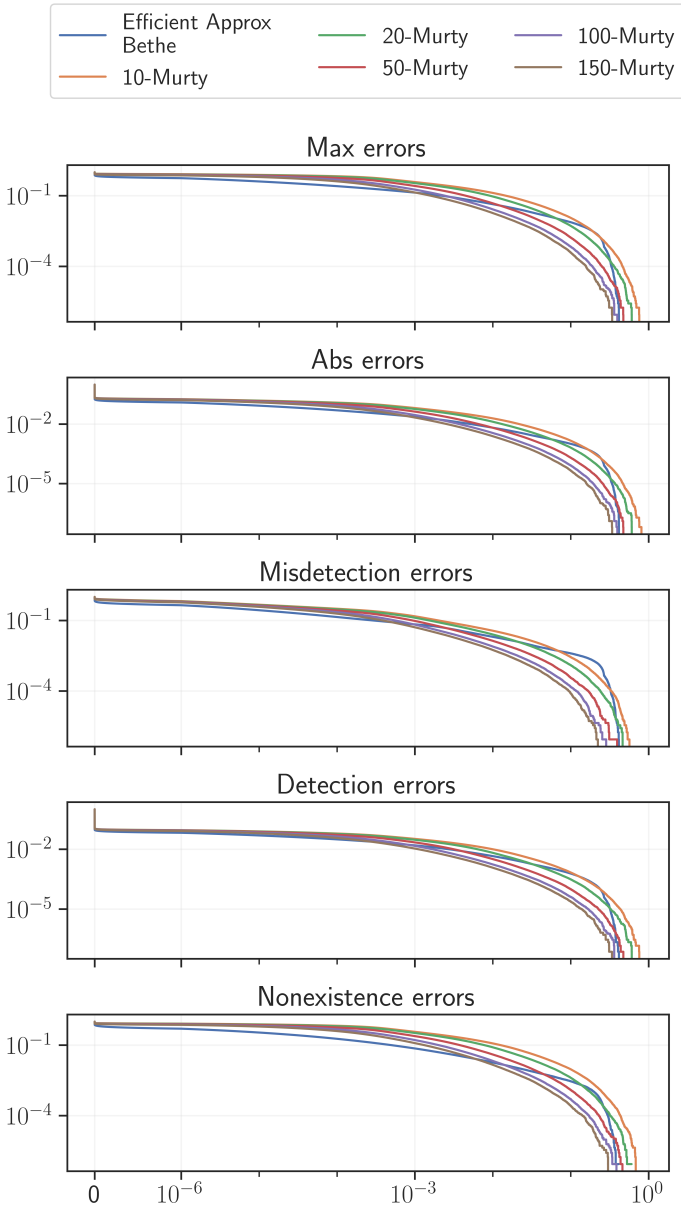


Figure 10.7: *Survival function plots over marginal errors comparing "Efficient Approx Bethe" to the different configurations of Murty's. All plots show the absolute value of the marginal errors. The "Max errors" plot is over the maximum error for each estimated track marginal distribution. The "Abs errors" plot contains all errors. The "Misdetection errors" plot contains only the misdetection errors, and similarly for "Detection errors" and "Nonexistence errors". The symlog scaling is used on the x-axis, with a linear threshold of* 10^{-6}*. The y-axis is logscale. The different configurations of Murty, i.e. the maximum number of hypotheses that Murty's is allowed to enumerate, is indicated by* K *in* K*-Murty.*

10.6 Convergence results for MCMHLBP

To finish we will briefly present results concerning convergence of MCMH-LBP. An analysis of a single case where the messages started to oscillate in MH-LBP can be found in [1], and it is reasonable to assume the same mechanics hold for MCMH-LBP.

Denote a metric that measures convergence by ε such that when $\varepsilon < \Delta$ for some threshold Δ, we terminate. To investigate different metrics to measure convergence, two different metrics were used. The first is the message norm from [8], specifically in our implementation

$$\varepsilon_M = d(\nu^k, \nu^{k-1}) \tag{10.1}$$

$$= \max_{i,j} \left| \log \frac{\nu^k_{i\to j}}{\nu^{k-1}_{i\to j}} \right|, \tag{10.2}$$

where we use ν^k to indicate the set of measurement-to-track messages at iteration k and similarly for ν^{k-1}. We use this message in particular because it is the only message in the multi-cluster, multi-hypothesis factor graph that is defined the same as in [8]. It therefore has the same domain and range such that we can be certain that the values it takes are well-behaved and bounded, which is useful to check convergence. The other metric is the absolute difference of the Bethe pseudodual between two consecutive iterations,

$$\varepsilon_B = \left| F^{\#}_i - F^{\#}_{i-1} \right| \tag{10.3}$$

where $F^{\#}_i$ denotes the Bethe pseudodual in iteration i. The parameters used to determine convergence can be found in Table 10.1.

Parameter name	Symbol	Value
Maximum number iterations	$N_{\max}$	10000
Bethe pseudodual threshold	Δ_B	10^{-7}
Message norm threshold	Δ_M	10^{-5}

Table 10.1: *Convergence parameters used in the implemention of MCMHLBP.*

In only 2 cases out of all 9188 did MCMH-LBP not converge, which means that the number of iterations reach the maximum $N_{\max}$ before both $\varepsilon_M < \Delta_M$ and $\varepsilon_B < \Delta_B$ simultaneously. In Table 10.2 the value of the convergence metrics can be found when the algorithm terminated. In both cases, MCMH-LBP messages settled at oscillations. However, for the first failed case, it seems like MCMH-LBP was close to convergence,

as the distance between the messages is relatively small. Although the Bethe pseudodual strictly speaking is only equal to the Bethe free energy at a stationary point, we can for intuition consider the difference between two Bethe pseudodual values the *log-ratio of the corresponding normalization constants*. In other words, if we define the pseudodual constant $Z_i^{\#}$ such that

$$F_i^{\#} = -\ln Z_i^{\#} \tag{10.4}$$

then

$$\begin{aligned}
\varepsilon_B &= \left| F_i^{\#} - F_{i-1}^{\#} \right| && (10.5)\\
&= \left| -\ln Z_i^{\#} + \ln Z_{i-1}^{\#} \right| && (10.6)\\
&= \left| \ln \frac{Z_{i-1}^{\#}}{Z_i^{\#}} \right| && (10.7)\\
&= \left| \ln \frac{Z_i^{\#}}{Z_{i-1}^{\#}} \right|. && (10.8)
\end{aligned}$$

Thus, for the first failed case, the ratio in (10.8) is $\exp(0.00500995) \approx 1.005$, and so we expect at least the normalization constant to have converged to some value. Doing the same for the message norm we get $\exp(0.15219049) \approx 1.164$, which seems to indicate that the marginals also have converged toward some value. This is, however, harder to say since the marginals depend on all the messages in the graph. Doing the same for the second case gives a normalization constant ratio of approximately 1.49 and message norm ratio of 60.937, which indicates that this case is much farther from convergence than the first one, since the ratios are much larger.

	Failed case 1	Failed case 2
Bethe pseudodual error ε_B	0.00500995	0.4000705
Message norm ε_M	0.15219049	4.109838

Table 10.2: *The value of the convergence metrics at the two cases where MCMHLBP failed to converge.*

For the cases where MCMH-LBP converged, a plot over the number of iterations until convergence and the corresponding convergence metric value at convergence can be found in Figure 10.8. Most cases seems to converge before 100 iterations, considerably less than the threshold of 10000. Perhaps the most interesting observation we can make is that the Bethe pseudodual converges faster than the message norm despite having a stricter threshold, as it converges in less iterations than the message norm.

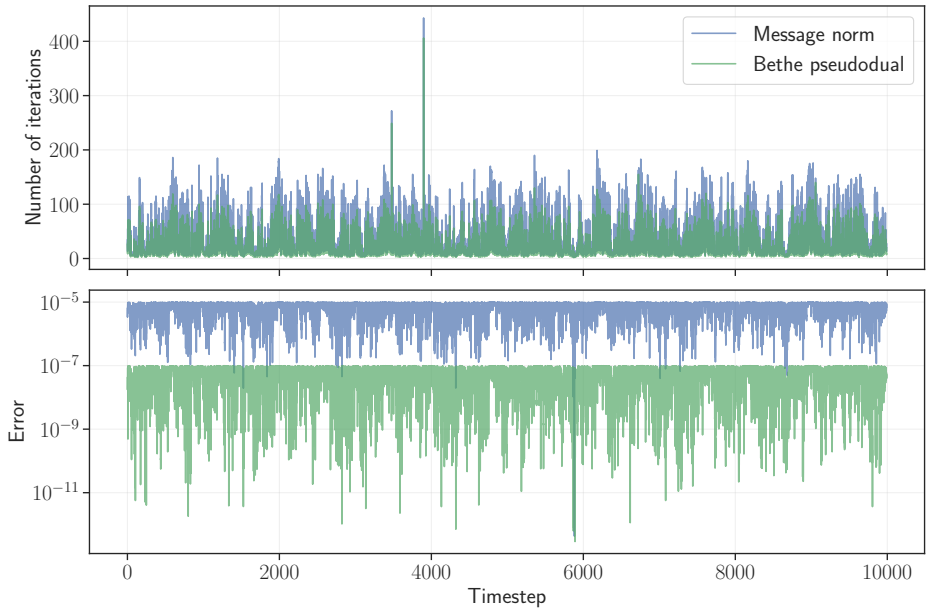


Figure 10.8: *The top plot shows number of iterations until convergence for the two convergence metrics used while the bottom plot shows the value of the metric at convergence. Note that only convergent cases are plotted. Additionally, the y-axis on the bottom plot is logscale. Lastly, the legend is the same for both plots.*

We end our discussion on the convergence of MCMH-LBP here. Future work is to further delve into the accuracy of MCMH-LBP marginals and normalization constant when only using one of the metrics and for different thresholds. It would seem from our results that the normalization constant estimate converges faster than the messages and therefore the estimated marginals. This is a useful observation if only a normalization constant estimate is wanted, as the algorithm can be terminated earlier without loss of accuracy.

IV

Closing remarks

11 | Conclusion

The main contributions presented in this thesis are different methods for efficient marginalization and estimation of the normalization constant of the multi-cluster, multi-hypothesis association graph. This was mainly motivated to enable track recycling in PMBM, which will allow for a more sparse MBM component with less hypotheses while still maintaining track cardinality balance. Four approaches were presented and tested that all build upon the LBP algorithm. Additionally, novel equations based on the specialized LBP messages were presented that uses the Bethe pseudodual to compute the Bethe constant as an estimate for the true normalization constant.

The first approach uses LBP on the full association graph and was called MCMH-LBP. The three other methods embeds LBP into a novel cluster-conditioning marginalization method that avoids prior hypothesis enumeration by conditioning on the linking measurements between prior clusters. One of the methods, called MH-LBP, uses LBP on the resulting single-cluster, multi-hypothesis association graphs. The two last methods, "Efficient Approx Bethe" and "Efficient Approx PHD", further marginalizes over the prior hypotheses to perform LBP on single-cluster, single-hypothesis association graphs. The two methods differs in how the hypothesis-conditioned likelihood is estimated, where "Efficient Approx Bethe" uses the Bethe pseudodual and "Efficient Approx PHD" uses a Poisson approximation of the likelihood based on ideas from the PHD filter.

The MCMH-LBP method was tested on a simple test case where the data association parameters were manipulated to explore the dynamics of LBP in a multi-hypothesis setting. The main observation was that the accuracy of the estimated marginals and normalization constant seems to be related to the track distribution across prior hypotheses.

All four methods were tested on a large dataset and compared with a benchmark based on Murty's method. From the results, we saw that MCMH-LBP estimated the normalization constant significantly better than the other presented methods. This illustrated the accuracy of the Bethe constant to estimate the true normalization constant. The other methods showed a much larger variance, which we explained as coming from the

inclusion-exclusion approximation that is made. The "Efficient Approx Bethe" method estimated the association marginals considerably better than the other methods, except for the misdetection probability. We explained this by the overcounting of the false alarm event that is done by the inclusion-exclusion approximation.

A consistent pattern was observed in the marginal errors from the multi-hypothesis LBP methods that does not appear in the hypothesis-conditioned methods. To inspect the reason for this pattern, the prior hypothesis posterior estimates of the different methods were compared. It was then revealed that multi-hypothesis LBP has a tendency to overestimate the posterior probabilities, while the hypothesis-conditioned methods does not. This led to the conclusion that hypothesis-conditioned LBP is a more reliable estimator for marginals in the general case.

11.1 Future work

There are several topics to consider for future work in this thesis. The results from "Efficient Approx Bethe" suggested that a large source of error in the cluster-conditioning methods is from the inclusion-exclusion approximation made to get more stable LBP estimates. Therefore, one should make a potentially suboptimal implementation that computes this sum exactly to verify whether this indeed is the case. If so, developing a method for computing the inclusion-exclusion sum efficiently and exactly should be possible given the structure of the problem, which would greatly improve the accuracy.

The motivation for developing the presented methods was to enable track recycling in PMBM to improve filter consistency. This should be verified by integrating the methods presented into an actual PMBM implementation. In particular, one should compare the filter output from a PMBM filter with track recycling against an implementation that uses Murty's method internally to properly test whether Murty's method actually is better and more reliable.

Another important motivation for the presented methods was the efficiency improvements they provide. However, due to implementation details, this comparison was never made in the present work. Implementing the presented methods in a way that makes comparing e.g. runtime fair should therefore also be done.

There are multiple, useful theoretical results about the single-cluster, single-hypothesis LBP method that makes it favorable for reliability. However, no such theoretical results exist for the generalized MH-LBP and MCMH-LBP methods. Based on the results in Chapters 8 and 10 it would seem like multi-hypothesis LBP, in particular the multi-hypothesis Bethe free energy, does not exhibit the same favorable properties as the

single-hypothesis equivalent. Because of this, at least three different routes forward are possible.

Firstly, it might be possible to improve the multi-hypothesis LBP methods by instead taking the approach in e.g. [30] where they used fractional free energy to improve the inference accuracy.

Secondly, exploring different inference methods on the multi-hypothesis association graph, e.g. the closely related variation inference method *mean field approximation* [50], can be useful.

Lastly, one can consider delving deeper into the theoretical aspect of the multi-hypothesis Bethe free energy. This can shed light on the dynamics of multi-hypothesis LBP and reveal its failure modes.

V

Appendices

A | Derivation of MH-LBP messages

Due to the similarities between the single-cluster and multi-cluster association graphs, the following section will first prove in Lemma 1 the single-cluster, multi-hypothesis LBP messages that was proved in the preceding project report, for then to argue about how they generalize to the multi-cluster case in the end.

Lemma 1 (The message definitions for single-cluster, multi-hypothesis LBP)**.** *Given an association graph of the same structure as in Figure 5.1 where the factors are defined as in* (5.12) *to* (5.15)*, the normalized messages used in single-cluster, multi-hypothesis LBP are given as*

$$\mu_{t\to j} = \frac{\psi^t(j)}{\psi^t(0) + \sum_{j'\neq j, j'>0} \psi^t(j')\nu_{j'\to t} + \sigma_t}, \tag{A.1a}$$

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

$$\sigma_t = \rho_t \cdot \frac{\sum_{\theta\,:\,t\notin\theta} \varphi(\theta) \prod_{t'\in\theta} \rho_{t'}}{\sum_{\theta\,:\,t\in\theta} \varphi(\theta) \prod_{t'\in\theta} \rho_{t'}}, \tag{A.1c}$$

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

where $\sum_{j'\neq j, j'>0}$ denotes the sum over all values $j' = 1, \ldots, m_k$ except for j for m_k measurements, $\sum_{t'\neq t, t'>0}$ denotes the sum over all values $t' = 1, \ldots, n_k$ except for t for n_k tracks, $\sum_{\theta\,:\,t\in\theta}$ denotes the sum over all prior hypotheses θ where track t exists and vice versa for $\sum_{\theta\,:\,t_c\notin\theta}$ and $\prod_{t'\in\theta^c}$ denotes the product over all tracks t that exist in the prior hypothesis θ.

Proof. The following proof is taken from the preceding project report of this thesis and is original work by the author. In principle, doing LBP is matter of computing the

messages

$$\mu_{a\to i}(x_i) \leftarrow \sum_{x_{\mathsf{N}(a)\setminus\{i\}}} f_a(x_{\mathsf{N}(a)}) \prod_{j\in\mathsf{N}(a)\setminus\{i\}} \mu_{j\to a}(x_j), \tag{A.2}$$

$$\mu_{i\to a}(x_i) \leftarrow \prod_{b\in\mathsf{N}(i)\setminus\{a\}} \mu_{b\to i}(x_i) \tag{A.3}$$

repeatedly until convergence, where the equations (A.2) and (A.3) are the same as in (2.15) and (2.16) and repeated here for convenience. 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 A.1 and their directions illustrated in Figure A.1.

By inserting the factors (5.12) to (5.15) that we defined in Chapter 5 into (A.2) and (A.3) and using the message notation from Table A.1, the general LBP equations take the form

$$\mu_{t\to j}(b^j) = \sum_{a^t} \psi^t(a^t)\gamma_{tj}(a^t, b^j) \left(\prod_{j'\neq j} \nu_{j'\to t}(a^t) \right) \sigma_t(a^i), \tag{A.4}$$

$$\nu_{j\to t}(a^t) = \sum_{b^j} \gamma^{jt}(a^t, b^j) \prod_{t'\neq t} \mu_{t'\to j}(b^j), \tag{A.5}$$

$$\sigma_t(a^t) = \sum_{\theta} \zeta^t(\theta, a^t)\varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta), \tag{A.6}$$

$$\rho_t(\theta) = \sum_{a^t} \zeta^t(\theta, a^t)\psi_t(a^t) \prod_{j} \nu_j(a^t), \tag{A.7}$$

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.

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

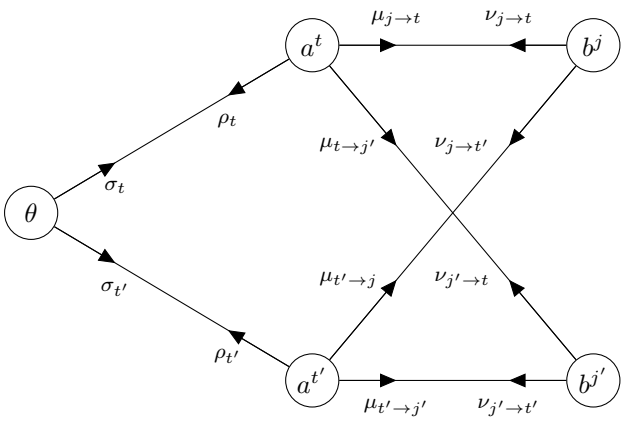


Figure A.1: *Simplified illustraion of message directions in association graph.*

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 A.1: *Message types in association graph.*

the sum into the term where $a^t = j$ and a partial sum over the remaining a^t as

$$\mu_{t\to j}(b^j) = \psi_t(a^t = j)\gamma_{tj}(a^t = j, b^j)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t)\right)\sigma_t(a^t) + \sum_{a^t\neq j}\psi_t(a^t)\gamma_{tj}(a^t \neq j, b^j)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t)\right)\sigma_t(a^t). \tag{A.8}$$

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

$$\sum_{a^t\neq j}\psi_t(a^t)\gamma_{tj}(a^t \neq j, b^j = t)\prod_{j'\neq j}\nu_{j'\to t} = 0 \tag{A.9}$$

and

$$\psi_t(a^t = j)\gamma_{tj}(a^t = j, b^j)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t)\right)\sigma_t(a^t) = \psi_t(a^t = j)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t)\right)\sigma_t(a^t) \tag{A.10}$$

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 (5.15). Doing the same for $b^j \neq t$ gives

$$\psi_t(a^t = j)\gamma_{tj}(a^t = j, b^j \neq t) \prod_{j' \neq j} \nu_{j' \to t}(a^t)\sigma_t(a^t) = 0 \tag{A.11}$$

and

$$\sum_{a^t \neq j} \psi_t(a^t)\gamma_{tj}(a^t \neq j, b^j = t) \prod_{j' \neq j} \nu_{j' \to t} = \sum_{a^t \neq j} \psi_t(a^t) \prod_{j' \neq j} \nu_{j' \to t} \tag{A.12}$$

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

$$\mu_{t \to j}(b^j) = \begin{cases} \psi_t(a^t = j) \left(\prod_{j' \neq j} \nu_{j' \to t}(a^t = j) \right) \sigma_t(a^t = j), & b^j = t \\ \sum_{a^t \neq j} \psi_t(a^t) \prod_{j' \neq j} \nu_{j' \to t}(a^t)\sigma_t(a^t), & b^j \neq t. \end{cases} \tag{A.13}$$

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

$$\mu_{t \to j}(b^j = t) = \frac{\psi_t(a^t = j) \prod_{j' \neq j} \nu_{j' \to t}(a^t)\sigma_t(a^t)}{\sum_{a^t \neq j} \psi_t(a^t) \prod_{j' \neq j} \nu_{j' \to t}(a^t)\sigma_t(a^t)}, \tag{A.14}$$

$$\mu_{t \to j}(b^j \neq t) = 1. \tag{A.15}$$

For now, these are the simplifications we can do. The expression in (A.14) 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

$$\begin{aligned} \nu_{j \to t}(a^t) = \gamma_{tj}(a^t, b^j = t) \prod_{t' \neq t} \mu_{t' \to j}(b^j = t) \\ + \sum_{b^t \neq t} \gamma_{tj}(a^t, b^j) \prod_{t' \neq t} \mu_{t' \to j}(b^j). \end{aligned} \tag{A.16}$$

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

$$\nu_{j \to t}(a^t) = \begin{cases} \prod_{t' \neq t} \mu_{t' \to j}(b^j = t), & a^t = j \\ \sum_{b^t \neq t} \prod_{t' \neq t} \mu_{t' \to j}(b^j), & a^t \neq j. \end{cases} \tag{A.17}$$

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

$$\nu_{j\to t}(a^t = j) = \frac{\prod_{t'\neq t} \mu_{t'\to j}(b^j = t)}{\sum_{b^t\neq t} \prod_{t'\neq t} \mu_{t'\to j}(b^j \neq t)}, \tag{A.18}$$

$$\nu_{j\to t}(a^t \neq j) = 1. \tag{A.19}$$

If we now insert (A.15) into (A.18) we get that the numerator reduces to

$$\prod_{t'\neq t} \mu_{t'\to j}(b^j = t) = \prod_{t'\neq t} 1 \tag{A.20}$$

$$= 1 \tag{A.21}$$

and the denominator becomes

$$\sum_{b^t\neq t} \prod_{t'\neq t} \mu_{t'\to j}(b^j \neq t) = \prod_{t'\neq t} \mu_{t'\to j}(b^j = 0) + \sum_{\substack{t''>0\\ t''\neq t}} \prod_{t'\neq t} \mu_{t'\to j}(b^j = t'') \tag{A.22}$$

$$= \prod_{t'\neq t} 1 + \sum_{\substack{t''>0\\ t''\neq t}} \left(\mu_{t''\to j}(b^j = t'') \prod_{\substack{t'\neq t\\ t'\neq t''}} 1 \right) \tag{A.23}$$

$$= 1 + \sum_{\substack{t''>0\\ t''\neq t}} \mu_{t''\to j}(b^j = t''), \tag{A.24}$$

which, after changing back the dummy variable t'' to t' in (A.24), gives the final expression

$$\nu_{j\to t} = \frac{1}{1 + \sum_{t'\neq t, t'>0} \mu_{t'\to j}}, \tag{A.25}$$

which is the same as in (6.1b).

Next we turn to the hypothesis-to-track message σ_t. If we first rewrite the sum in (A.6) as the sum of two partial sums,

$$\sigma_t(a^t) = \sum_{\theta:\, t\in\theta} \varphi(\theta)\zeta_t(\theta, a^t) \prod_{t'\neq t} \rho_{t'}(\theta) + \sum_{\theta:\, t\notin\theta} \varphi(\theta)\zeta_t(\theta, a^t) \prod_{t'\neq t} \rho_{t'}(\theta), \tag{A.26}$$

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 $t \in \theta$

$$\sum_{\theta:\, t \notin \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) = 0 \tag{A.27}$$

and

$$\sum_{\theta:\, t \in \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) = \sum_{\theta:\, t \in \theta} \varphi(\theta) \prod_{t' \neq t} \rho_{t'}(\theta) \tag{A.28}$$

as $\zeta_t(\theta, a^t) = 0$ and $\zeta_t(\theta, a^t) = 1$, respectively, by the way it was defined in (5.13). Similarly, when $a^t = N$ and $t \in \theta$,

$$\sum_{\theta:\, t \in \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) = 0 \tag{A.29}$$

and

$$\sum_{\theta:\, t \notin \theta} \varphi(\theta) \zeta_t(\theta, a^t) \prod_{t' \neq t} \rho_{t'}(\theta) = \sum_{\theta:\, t \notin \theta} \varphi(\theta) \prod_{t' \neq t} \rho_{t'}(\theta), \tag{A.30}$$

Thus, σ_t reduces to the two cases

$$\sigma_t(a^t) = \begin{cases} \sum_{\theta:\, t \in \theta} \varphi(\theta) \prod_{t' \neq t} \rho_{t'}(\theta), & a^t = 0, 1, \ldots, m_k \\ \sum_{\theta:\, t \notin \theta} \varphi(\theta) \prod_{t' \neq t} \rho_{t'}(\theta), & a^t = N. \end{cases} \tag{A.31}$$

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

$$\sigma_t(a^t = N) = \frac{\displaystyle\sum_{\theta:\, t \notin \theta} \varphi(\theta) \prod_{t' \neq t} \rho_{t'}(\theta)}{\displaystyle\sum_{\theta:\, t \in \theta} \varphi(\theta) \prod_{t' \neq t} \rho_{t'}(\theta)}, \tag{A.32}$$

$$\sigma_t(a^t \neq N) = 1. \tag{A.33}$$

We will return to (A.32) 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 (A.14). Inserting (A.19) and (A.33) into (A.14) makes the numerator

$$\psi_t(a^t = j) \left(\prod_{j' \neq j} \nu_{j' \to t}(a^t = j) \right) \sigma_t(a^t = j) = \psi_t(a^t = j) \left(\prod_{j' \neq j} 1 \right) \cdot 1 \tag{A.34}$$

$$= \psi_t(a^t = j) \tag{A.35}$$

and the denominator

$$\begin{aligned}\sum_{a^t\neq j}\psi^t(a^t)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t)\right)\sigma_t(a^t) &= \psi^t(a^t=0)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t=0)\right)\sigma_t(a^t=0)\\ &\quad+\sum_{\substack{a^t=1\\ a^t\neq j}}^{m_k}\psi^t(a^t)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t)\right)\sigma_t(a^t)\\ &\quad+\psi^t(a^t=N)\left(\prod_{j'\neq j}\nu_{j'\to t}(a^t=N)\right)\sigma_t(a^t=N) \qquad \text{(A.36)}\\ &= \psi^t(a^t=0)\left(\prod_{j'\neq j}1\right)\cdot 1\\ &\quad+\sum_{\substack{a^t=1\\ a^t\neq j}}^{m_k}\psi^t(a^t)\nu_{a^t\to t}(a^t)\left(\prod_{\substack{j'\neq j\\ j'\neq a^t}}1\right)\cdot 1\\ &\quad+1\cdot\left(\prod_{j'\neq j}1\right)\sigma_t(a^t=N) \qquad \text{(A.37)}\\ &= \psi^t(0)+\sum_{j'\neq j,j'>0}\psi^t(j')\nu_{j'\to t}+\sigma_t \qquad \text{(A.38)}\end{aligned}$$

where we used that $\psi^t(a^t=N)=1$ from (5.14c). Putting it back together we get

$$\mu_{t\to j}=\frac{\psi_t(j)}{\psi_t(0)+\sum_{j'\neq j,j'>0}\psi_t(j')\nu_{j'\to t}+\sigma_t}. \qquad \text{(A.39)}$$

which again is the desired result in (6.1a).

The track-to-hypothesis message $\rho_t(\theta)$ can be simplified as follows. We do a decomposition of the sum in (A.7) into a partial sum over $a^t=0,1,\ldots,m_k$ and the term for $a^t=N$ to get

$$\rho_t(\theta)=\sum_{a^t\neq N}\psi^t(a^t)\zeta^t(\theta,a^t)\prod_j\nu_{j\to t}(a^t)+\psi^t(N)\zeta^t(\theta,a^t=N)\prod_j\nu_{j\to t}(N). \qquad \text{(A.40)}$$

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

$$\psi^t(N)\zeta^t(\theta, a^t = N)\prod_j \nu_{j\to t}(N) = 0 \tag{A.41}$$

and

$$\sum_{a^t \neq N} \psi^t(a^t)\zeta^t(\theta, a^t)\prod_j \nu_{j\to t}(a^t) = \sum_{a^t \neq N} \psi^t(a^t)\prod_j \nu_{j\to t}(a^t) \tag{A.42}$$

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

$$\sum_{a^t \neq N} \psi^t(a^t)\zeta^t(\theta, a^t)\prod_j \nu_{j\to t}(a^t) = 0 \tag{A.43}$$

and

$$\psi^t(N)\zeta^t(\theta, a^t = N)\prod_j \nu_{j\to t}(N) = \psi^t(N)\prod_j \nu_{j\to t}(N) \tag{A.44}$$

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

$$\rho_t(\theta) = \begin{cases} \sum_{a^t \neq N} \psi_t(a^t)\prod_j \nu_j(a^t), & t \in \theta \\ \psi_t(N)\prod_j \nu_j(N). & t \notin \theta \end{cases} \tag{A.45}$$

By inserting $\psi^t(N) = 1$ from (5.14c) and $\prod_j \nu_j(N) = 1$ from (A.19) 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

$$\rho_t = \psi^t(0) + \sum_{j=1}^{m_k} \psi^t(j)\nu_{j\to t} \tag{A.46}$$

which we recognize as (6.1d).

The only thing that remains is to simplify (A.32). 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

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

$$= \rho_t \cdot \frac{\sum\limits_{\theta:\, t\notin\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}{\sum\limits_{\theta:\, t\in\theta} \varphi(\theta) \prod\limits_{t'\in\theta} \rho_{t'}}. \tag{A.47}$$

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. ■

With Lemma 1, it is straight-forward to prove Theorem 1.

Proof. To visualize the necessary changes, see Figure A.2, which is the same figure as Figure 6.1. The most important thing to realize is that locally to each node, the edges look the same, and so we expect the expressions to be similar. The track-to-measurement messages $\mu_{t\to j}$ and measurement-to-track messages $\nu_{j\to t}$ remain the same as a track a^t only has one incoming edge from the corresponding hypothesis variable θ^c and the edges from messages b^j. The edges into each measurement b^j are equal to that of the single-cluster graph and so the equality follows trivially. The same then also holds for the track-to-hypothesis messages ρ_t as they are entirely a function of $\nu_{j\to t}$ and the prior factors $\psi^t(j)$, where the prior factors also are the same. The only major difference are the hypothesis-to-track messages σ_{t_c} which has to be computed for each cluster. The expression becomes the same, although we now need specify what tracks t_c to use as the incoming edges to θ^c are only from tracks a^{t_c} that participates in the given cluster c.

■

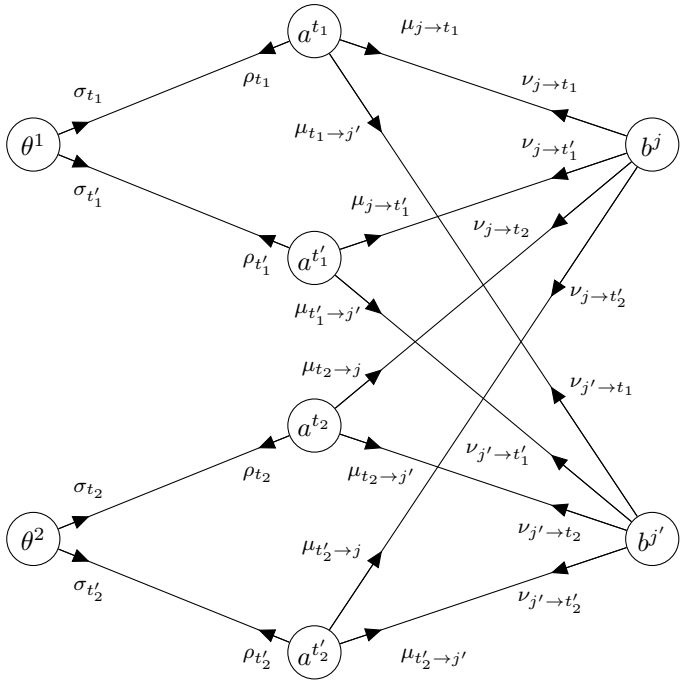


Figure A.2: *Message direction example for multicluster scenario with two clusters.*

B | Derivation of MH-LBP Bethe pseudodual

In the following we will adhere to the pairwise potential Markov random field equivalent of the MH factor graph. In other words, the MH association density $p(\boldsymbol{x})$ is given as

$$p(\boldsymbol{x}) \propto \prod_{i \in \mathcal{V}} \phi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \tag{B.1}$$

where $\mathcal{V}$ denotes the index set for the vertex indices and $\mathcal{E}$ the index set for the edge indices of the graph. We rewrite (B.1) as

$$p(\boldsymbol{x}) \propto \prod_{i \in \mathcal{V}} \exp\left\{\tilde{\phi}_i(x_i)\right\} \prod_{(i,j) \in \mathcal{E}} \exp\left\{\tilde{\psi}_{ij}\right\}(x_i, x_j) \tag{B.2}$$

where

$$\tilde{\phi}_i(x_i) = \ln \phi_i(x_i)) \tag{B.3}$$

$$\tilde{\psi}_{ij}(x_i, x_j) = \ln \psi_{ij}(x_i, x_j) \tag{B.4}$$

We will later see that the case $\ln 0$ does not need to be properly handled. Recall that the Bethe free energy is defined as

$$F_B = U_B - H_B \tag{B.5}$$

where

$$U_B = \mathbb{E}_q \left[E(\boldsymbol{x})\right], \tag{B.6}$$

$$H_B = -\mathbb{E}_q \left[\ln q(\boldsymbol{x})\right] \tag{B.7}$$

where the Bethe approximation $q(\boldsymbol{x})$ factors as

$$q(\boldsymbol{x}) \propto \prod_{i \in \mathcal{V}} q_i(x_i) \prod_{(i,j) \in \mathcal{E}} \frac{q_{ij}(x_i, x_j)}{q_i(x_i) q_j(x_j)}. \tag{B.8}$$

In our case, the beliefs q_i and q_{ij} are given by

$$q_\theta(\theta) = \frac{1}{Z_\theta} \varphi(\theta) \prod_t \rho_t(\theta), \tag{B.9}$$

$$q_t(a^t) = \frac{1}{Z_t} \psi^t(a^t) \prod_j \nu_{j \to t}(a^t) \sigma_t(a^t), \tag{B.10}$$

$$q_j(b^j) = \frac{1}{Z_j} \prod_t \mu_{t \to j}(b^j), \tag{B.11}$$

$$q_{t\theta}(a^t, \theta) = \frac{1}{Z_{t\theta}} \psi^t(a^t) \zeta^t(\theta, a^t) \varphi(\theta) \prod_j \nu_{j \to t}(a^t) \prod_{t' \neq t} \rho_{t'}(\theta), \tag{B.12}$$

$$q_{tj}(a^t, b^j) = \frac{1}{Z_{tj}} \psi^t(a^t) \gamma^{tj}(a^t, b^j) \prod_{j' \neq j} \nu_{j' \to t}(a^t) \prod_{t' \neq t} \mu_{t' \to j}(b^j) \sigma_t(a^t). \tag{B.13}$$

where q_θ, q_t, and q_j are the node beliefs for θ, a^t and b^j, respectively, and $q_{t\theta}$ and q_{tj} are the edge beliefs for a^t and θ and for a^t and b^j, respectively.

Inserting (B.8) into (B.5) and rewriting the resulting expression based on what is done in [51], [52], we arrive at the expression

$$F_B = \sum_{i \in \mathcal{V}} (1 - d_i) \sum_{x_i \in \mathcal{X}} \left[\ln q_i(x_i) - \tilde{\phi}_i(x_i) \right] \tag{B.14}$$

$$+ \sum_{(i,j) \in \mathcal{E}} \sum_{x_i, x_j \in \mathcal{X}^2} \left[\ln q_{ij}(x_i, x_j) - \tilde{\phi}_i(x_i) - \tilde{\phi}_j(x_j) - \tilde{\psi}_{ij}(x_i, x_j) \right] \tag{B.15}$$

where d_i denotes the degree, i.e. the number of adjacent edges, of vertex i.

In [26] they argue that one should use the *pseudodual* of the Bethe free energy function, as it allows for tracking convergence of LBP while also evaluating to exactly the Bethe free energy function at the stationary point of the LBP messages. The pseudodual is constructed as a modified Lagrangian by simply adding the edge constraints in the

original LBP optimization problem, such that the pseudodual $F_B^{\#}$ takes the form

$$F_B^{\#} = F_B$$
$$- \sum_{(i,j)\in\mathcal{E}} \sum_{x_i\in\mathcal{X}} \lambda_{j\to i}(x_i) \left(\sum_{x_j\in\mathcal{X}} q_{ij}(x_i,x_j) - q_i(x_i) \right) \tag{B.16}$$
$$- \sum_{(i,j)\in\mathcal{E}} \sum_{x_j\in\mathcal{X}} \lambda_{i\to j}(x_j) \left(\sum_{x_i\in\mathcal{X}} q_{ij}(x_i,x_j) - q_j(x_j) \right). \tag{B.17}$$

where the Lagrange multipliers $\lambda_{i\to j}$ and $\lambda_{j\to i}$ are given from the messages in LBP as

$$\lambda_{i\to j} = \sum_{k\in\mathsf{N}(j)\setminus\{i\}} \ln m_{k\to j}(x_j) \tag{B.18}$$

and where $m_{i\to j}$ denotes an LBP message from node i to j and $\mathsf{N}(i)$ denotes the neighbors of node i as proven in [23] and the notation is from [51].

Initially, the above expression looks more complicated than just the Bethe free energy function. We will, however, see that evaluating the pseudodual is in general simpler than the Bethe free energy function due to several beneficial cancellations of terms.

The proof will first expand the node and edge terms with the relevant factors and simplify as much as possible. Then, the constraint terms will be expanded, for lastly to go over all cancellations that occur before concluding the proof. To help the derivation we will introduce the notation

$$F_B^{\#} = F_n + F_e - L \tag{B.19}$$

where F_n denotes the sums over nodes, F_e the sums over edges and L the constraint sums.

The node terms in (B.14) can be expanded as follows. The node term for θ we get $(1-n)\sum_\theta q_\theta(\theta)\left(\ln q_\theta - \ln\varphi(\theta)\right)$, which is reduced to

$$(1-n)\sum_\theta q_\theta(\theta)\left(\ln q_\theta - \ln\varphi(\theta)\right) = (1-n)\sum_\theta q_\theta(\theta)\left(-\ln Z_\theta + \sum_{t=1}^{n} \ln\rho_t\right) \tag{B.20}$$

by substituting q_θ with (B.9) and canceling. Similarly, for the track nodes and measurement nodes we get $\sum_{t=1}^{n}(-m)\sum_{a^t} q_t(a^t)\left(\ln q_t - \ln\psi^t(a^t)\right)$ and $\sum_{t=1}^{n}(1-n)\sum_{b^j} q_j(b^j)\ln q_j$

which are reduced to

$$\sum_{t=1}^{n}(-m)\sum_{a^t} q_t(a^t)\left(\ln q_t - \ln \psi^t(a^t)\right) = \sum_{t=1}^{n}(-m)\sum_{a^t} q_t(a^t)\left(-\ln Z_t + \ln \sigma_t + \sum_{j=1}^{m}\ln \nu_{j\to t}\right), \tag{B.21}$$

$$\sum_{t=1}^{n}(1-n)\sum_{b^j} q_j(b^j)\ln q_j = \sum_{t=1}^{n}(1-n)\sum_{b^j} q_j(b^j)\left(-\ln Z_j + \sum_{t=1}^{m}\ln \mu_{t\to j}\right). \tag{B.22}$$

In conclusion, we can write F_n as

$$F_n = (1-n)\sum_{\theta} q_\theta(\theta)\left(-\ln Z_\theta + \sum_{t=1}^{n}\ln \rho_t\right) \tag{B.23a}$$

$$+\sum_{t=1}^{n}(-m)\sum_{a^t} q_t(a^t)\left(-\ln Z_t + \ln \sigma_t + \sum_{j=1}^{m}\ln \nu_{j\to t}\right) \tag{B.23b}$$

$$+\sum_{t=1}^{n}(1-n)\sum_{b^j} q_j(b^j)\left(-\ln Z_j + \sum_{t=1}^{m}\ln \mu_{t\to j}\right) \tag{B.23c}$$

The edge terms takes the forms $\sum_{t=1}^{n}\sum_{a^t}\sum_{\theta} q_{t\theta}(a^t,\theta)\left(-\ln Z_{t\theta} + \sum_{j=1}^{m}\nu_{j\to t} + \sum_{t'\neq t}\rho_{t'}\right)$ for edges between θ and a track a^t and

$\sum_{t=1}^{n}\sum_{j=1}^{m}\sum_{a^t}\sum_{b^j} q_{tj}(a^t,b^j)\left(-\ln Z_{tj} + \sum_{j'\neq j}\nu_{j'\to t} + \ln \sigma_t + \sum_{t'\neq t}\mu_{t'\to j}\right)$ between a track a^t and measurement b^j by simple substitution and cancelling of terms, and so F_e becomes

$$F_e = \sum_{t=1}^{n}\sum_{a^t}\sum_{\theta} q_{t\theta}(a^t,\theta)\left(-\ln Z_{t\theta} + \sum_{j=1}^{m}\nu_{j\to t} + \sum_{t'\neq t}\rho_{t'}\right) \tag{B.24a}$$

$$+\sum_{t=1}^{n}\sum_{j=1}^{m}\sum_{a^t}\sum_{b^j} q_{tj}(a^t,b^j)\left(-\ln Z_{tj} + \sum_{j'\neq j}\nu_{j'\to t} + \ln \sigma_t + \sum_{t'\neq t}\mu_{t'\to j}\right). \tag{B.24b}$$

When expanding L, we separate the sum into messages going in both directions, and into the edges between θ and each a^t and similarly for between every a^t and b^j. We will first rename the Lagrangian multipliers with notation closer to its corresponding message counter-part. The hypothesis-to-track Lagrangian multiplier is called $\tilde{\sigma}_t(a^t)$ and is given

by

$$\tilde{\sigma}_t = \sum_{j=1}^{m} \ln \nu_{j\to t} \tag{B.25}$$

which follows from (B.18) and the structure of the graph. Similarly, the track-to-hypothesis Lagrangian multiplier is called $\tilde{\rho}_t$ and given by

$$\tilde{\rho}_t = \sum_{t'\neq t} \ln \rho_{t'}, \tag{B.26}$$

the track-to-measurement Lagrangian multiplier is called $\tilde{\mu}_{t\to j}$ and given by

$$\tilde{\mu}_{t\to j} = \sum_{t'\neq t} \ln \mu_{t'\to j} \tag{B.27}$$

and lastly, the measurement-to-track Lagrangian multiplier is called $\tilde{\nu}_{j\to t}$ and given by

$$\tilde{\nu}_{j\to t} = \sum_{j'\neq j} \ln \nu_{j'\to t} + \ln \sigma_t. \tag{B.28}$$

By expanding L with the Lagrangian multipliers and sorting the terms into edges between θ and a^t and between a^t and b^j, we get

$$\begin{aligned} L = \sum_{t=1}^{n} \Bigg\{ & \sum_{a^t} \tilde{\sigma}_t(a^t) \left(\sum_{\theta} q_{t\theta}(a^t,\theta) - q_t(a^t) \right) \\ & + \sum_{\theta} \tilde{\rho}_t(\theta) \left(\sum_{a^t} q_{t\theta}(a^t,\theta) - q_\theta(\theta) \right) \Bigg\} \qquad \text{(B.29a)} \\ & + \sum_{t=1}^{n} \sum_{j=1}^{m} \Bigg\{ \sum_{b^j} \tilde{\mu}_{t\to j}(b^j) \left(\sum_{a^t} q_{tj}(a^t,b^j) - q_j(b^j) \right) \\ & + \sum_{a^t} \tilde{\nu}_{j\to t}(a^t) \left(\sum_{b^j} q_{tj}(a^t,b^j) - q_t(a^t) \right) \Bigg\}. \qquad \text{(B.29b)} \end{aligned}$$

We proceed by writing out every sum in (B.29) individually while also substituting in the expressions for the Lagrangian multipliers found in (B.25) to (B.28) in addition to

making the algebraic manipulations

$$\sum_{t' \neq t} \ln \rho_{t'} = \sum_{t'=1}^{n} \ln \rho_{t'} - \ln \rho_t \tag{B.30}$$

$$\sum_{t' \neq t} \ln \mu_{t' \to j} = \sum_{t'=1}^{n} \ln \mu_{t' \to j} - \ln \mu_{t \to j} \tag{B.31}$$

$$\sum_{j' \neq j} \ln \nu_{j' \to t} = \sum_{j'=1}^{m} \ln \nu_{j' \to t} - \ln \nu_{j \to t} \tag{B.32}$$

in select places to arrive at the massive expression

$$L = \sum_{t=1}^{n} \sum_{a^t} \sum_{j=1}^{m} \ln \nu_{j \to t} \sum_{\theta} q_{t\theta}(a^t, \theta) \tag{B.33a}$$

$$- \sum_{t=1}^{n} \sum_{a^t} \sum_{j=1}^{m} \ln \nu_{j \to t} q_t(a^t) \tag{B.33b}$$

$$+ \sum_{t=1}^{n} \sum_{\theta} \sum_{t' \neq t} \ln \rho_{t'} \sum_{a^t} q_{t\theta}(a^t, \theta) \tag{B.33c}$$

$$- \sum_{t=1}^{n} \sum_{\theta} \sum_{t'=1}^{n} \ln \rho_{t'} q_\theta(\theta) \tag{B.33d}$$

$$+ \sum_{t=1}^{n} \sum_{\theta} \ln \rho_t q_\theta(\theta) \tag{B.33e}$$

$$+ \sum_{t=1}^{n} \sum_{j=1}^{m} \sum_{b^j} \sum_{t' \neq t} \ln \mu_{t' \to j} \sum_{a^t} q_{tj}(a^t, b^j) \tag{B.33f}$$

$$- \sum_{t=1}^{n} \sum_{j=1}^{m} \sum_{b^j} \sum_{t'=1}^{n} \ln \mu_{t' \to j} q_j(b^j) \tag{B.33g}$$

$$+ \sum_{t=1}^{n} \sum_{j=1}^{m} \sum_{b^j} \ln \mu_{t \to j} q_j(b^j) \tag{B.33h}$$

$$+ \sum_{t=1}^{n} \sum_{j=1}^{m} \sum_{a^t} \left(\sum_{j' \neq j} \ln \nu_{j' \to t} + \ln \sigma_t \right) \sum_{b^j} q_{tj}(a^t, b^j) \tag{B.33i}$$

$$- \sum_{t=1}^{n} \sum_{j=1}^{m} \sum_{a^t} \left(\sum_{j'=1}^{m} \ln \nu_{j' \to t} + \ln \sigma_t \right) q_t(a^t) \tag{B.33j}$$

$$+ \sum_{t=1}^{n} \sum_{j=1}^{m} \sum_{a^t} \ln \nu_{j \to t} q_t(a^t). \tag{B.33k}$$

What remains before cancelling terms is to collect sums in (B.33) to more easily recognize the cancellations we can make. Note that in (B.33d), (B.33g) and (B.33j) we repeat a sum twice, such that we can write the sums as

$$\sum_{t=1}^{n}\sum_{\theta}\sum_{t'=1}^{n}\ln\rho_{t'}q_\theta(\theta) = n\sum_{t=1}^{n}\sum_{\theta}\ln\rho_t q_\theta(\theta) \tag{B.34}$$

$$\sum_{t=1}^{n}\sum_{j=1}^{m}\sum_{b^j}\sum_{t'=1}^{n}\ln\mu_{t'\to j}q_j(b^j) = n\sum_{t=1}^{n}\sum_{j=1}^{m}\sum_{b^j}\ln\mu_{t\to j}q_j(b^j) \tag{B.35}$$

$$\sum_{t=1}^{n}\sum_{j=1}^{m}\sum_{a^t}\left(\sum_{j'=1}^{m}\ln\nu_{j'\to t}+\ln\sigma_t\right)q_t(a^t) = m\sum_{t=1}^{n}\sum_{a^t}\left(\sum_{j=1}^{m}\ln\nu_{j\to t}+\ln\sigma_t\right)q_t(a^t). \tag{B.36}$$

By combining (B.34) with (B.33e), (B.35) with (B.33h), (B.33a) with (B.33c), (B.33f) with (B.33i), cancelling (B.33b) with (B.33k), interchanging some sums and moving constants across sums we get that L can be written as

$$L = (1-n)\sum_{\theta}q_\theta(\theta)\sum_{t=1}^{n}\ln\rho_t \tag{B.37a}$$

$$+\sum_{t=1}^{n}(-m)\sum_{a^t}q_t(a^t)\left(\ln\sigma_t+\sum_{j=1}^{m}\ln\nu_{j\to t}\right) \tag{B.37b}$$

$$+\sum_{t=1}^{n}(1-n)\sum_{b^j}q_j(b^j)\sum_{t=1}^{m}\ln\mu_{t\to j} \tag{B.37c}$$

$$+\sum_{t=1}^{n}\sum_{a^t}\sum_{\theta}q_{t\theta}(a^t,\theta)\left(\sum_{j=1}^{m}\nu_{j\to t}+\sum_{t'\neq t}\rho_{t'}\right) \tag{B.37d}$$

$$+\sum_{t=1}^{n}\sum_{j=1}^{m}\sum_{a^t}\sum_{b^j}q_{tj}(a^t,b^j)\left(\sum_{j'\neq j}\nu_{j'\to t}+\ln\sigma_t+\sum_{t'\neq t}\mu_{t'\to j}\right). \tag{B.37e}$$

It is now trivial to make the necessary cancellations by comparing (B.37) with (B.23) and (B.24) while considering the signs in (B.19) to get that the Bethe pseudodual $F_B^{\#}$ can be written as

$$F_B^{\#} = (n-1)\ln Z_\theta + m\sum_{t=1}^{n}\ln Z_t + (n-1)\sum_{j=1}^{m}\ln Z_j - \sum_{t=1}^{n}\ln Z_{t\theta} - \sum_{t=1}^{n}\sum_{j=1}^{m}\ln Z_{tj}, \tag{B.38}$$

where we have marginalized out the LBP beliefs q_θ, q_t, q_j, $q_{t\theta}$ and q_{tj} as the only thing

remaining inside the relevant sums were the normalization constants Z_θ, Z_t, Z_j, $Z_{t\theta}$ and Z_{tj}.

We now derive how the normalization constants are computed by marginalizing the LBP beliefs, using the normalization properties of the messages involved and the compatibility factors ζ^t and γ^{tj}. For Z_θ we get

$$Z_\theta = \sum_\theta \varphi(\theta) \prod_t \rho_t(\theta) \tag{B.39}$$

$$= \sum_\theta \varphi(\theta) \prod_{t\in\theta} \rho_t \tag{B.40}$$

$$\tag{B.41}$$

with no more simplifications possible. For Z_t we get

$$Z_t = \sum_{a^t} \psi^t(a^t) \prod_j \nu_{j\to t}(a^t)\sigma_t(a^t) \tag{B.42}$$

$$= \psi^t(0) \prod_j \nu_{j\to t}(0)\sigma_t(0)$$

$$+ \sum_{j=1}^{m} \psi^t(j) \prod_j \nu_{j\to t}(j)\sigma_t(j)$$

$$+ \psi^t(N) \prod_j \nu_{j\to t}(N)\sigma_t(N) \tag{B.43}$$

$$= \psi^t(0) + \sum_{j=1}^{m} \psi^t(j)\nu_{j\to t} + \sigma_t. \tag{B.44}$$

For Z_j we get

$$Z_j = \sum_{b^j} \prod_t \mu_{t\to j}(b^j) \tag{B.45}$$

$$= \prod_t \mu_{t\to j}(0) + \sum_{t=1}^{n} \prod_{t'} \mu_{t\to j}(t') \tag{B.46}$$

$$= 1 + \sum_{t=1}^{n} \mu_{t\to j}. \tag{B.47}$$

The normalization constants for the edge beliefs are slightly more involved. For $Z_{t\theta}$ we

get

$$Z_{t\theta} = \sum_{\theta} \sum_{a^t} \psi^t(a^t)\zeta^t(\theta, a^t)\varphi(\theta) \prod_j \nu_{j\to t}(a^t) \prod_{t'\neq t} \rho_{t'}(\theta) \tag{B.48}$$

$$= \sum_{\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta) \sum_{a^t} \psi^t(a^t)\zeta^t(\theta, a^t) \prod_j \nu_{j\to t}(a^t) \tag{B.49}$$

$$= \sum_{\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta)\zeta^t(\theta, a^t \neq N) \left(\psi^t(0) \prod_j \nu_{j\to t}(0) + \sum_{j'=1}^{m} \psi^t(j') \prod_j \nu_{j\to t}(j') \right)$$
$$+ \sum_{\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta)\zeta^t(\theta, a^t = N)\psi^t(N) \prod_j \nu_{j\to t}(N) \tag{B.50}$$

$$= \sum_{\theta: t\in\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta) \left(\psi^t(0) \prod_j \nu_{j\to t}(0) + \sum_{j'=1}^{m} \psi^t(j') \prod_j \nu_{j\to t}(j') \right)$$
$$+ \sum_{\theta: t\notin\theta} \varphi(\theta) \prod_{t'\neq t} \rho_{t'}(\theta)\psi^t(N)\zeta^t(\theta, N) \prod_j \nu_{j\to t}(N) \tag{B.51}$$

$$= \frac{1}{\rho_t} \sum_{\theta: t\in\theta} \varphi(\theta) \prod_{t\in\theta} \rho_t \left(\psi^t(0) + \sum_{j=1}^{m} \psi^t(j)\nu_{j\to t} \right)$$
$$+ \sum_{\theta: t\notin\theta} \varphi(\theta) \prod_{t\in\theta} \rho_t\psi^t(N)$$
$$= \frac{\psi^t(0) + \sum_{j=1}^{m} \psi^t(j)\nu_{j\to t}}{\rho_t} \sum_{\theta: t\in\theta} \varphi(\theta) \prod_{t\in\theta} \rho_t + \sum_{\theta: t\notin\theta} \varphi(\theta) \prod_{t\in\theta} \rho_t. \tag{B.52}$$

Finally, for Z_{tj} we get

$$Z_{tj} = \sum_{a^t} \sum_{b^j} \psi^t(a^t)\gamma^{tj}(a^t, b^j) \prod_{j' \neq j} \nu_{j' \to t}(a^t) \prod_{t' \neq t} \mu_{t' \to j}(b^j)\sigma_t(a^t) \tag{B.53}$$

$$= \sum_{a^t \neq j} \psi^t(a^t) \prod_{j' \neq j} \nu_{j' \to t}(a^t)\sigma_t(a^t) \sum_{b^j \neq t} \prod_{t' \neq t} \mu_{t' \to j}(b^j)$$

$$+ \psi^t(j) \prod_{j' \neq j} \nu_{j' \to t}(j) \prod_{t' \neq t} \mu_{t' \to j}(t)\sigma_t(j) \tag{B.54}$$

$$= \psi^t(0) \prod_{j' \neq j} \nu_{j' \to t}(0)\sigma_t(0) \sum_{b^j \neq t} \prod_{t' \neq t} \mu_{t' \to j}(b^j)$$

$$+ \sum_{\substack{j''=1 \\ j'' \neq j}}^{m} \psi^t(j'') \prod_{j' \neq j} \nu_{j' \to t}(j'') \sum_{b^j \neq t} \prod_{t' \neq t} \mu_{t' \to j}(b^j)$$

$$+ \psi^t(N) \prod_{j' \neq j} \nu_{j' \to t}(N)\sigma_t(N) \sum_{b^j \neq t} \prod_{t' \neq t} \mu_{t' \to j}(b^j)$$

$$+ \psi^t(j) \tag{B.55}$$

$$= \left(\sum_{b^j \neq t} \prod_{t' \neq t} \mu_{t' \to j}(b^j) \right) \left(\psi^t(0) + \sum_{\substack{j'=1 \\ j' \neq j}}^{m} \psi^t(j')\nu_{j' \to t} + \psi^t(N)\sigma_t \right) + \psi^t(j) \tag{B.56}$$

$$= \left(1 + \sum_{\substack{t'=1 \\ t' \neq t}}^{n} \mu_{t' \to j} \right) \left(\psi^t(0) + \sum_{\substack{j'=1 \\ j' \neq j}}^{m} \psi^t(j')\nu_{j' \to t} + \sigma_t \right) + \psi^t(j) \tag{B.57}$$

C | Paper - “Belief propagation for marginal probabilities in multiple hypothesis tracking”

O. A. Severinsen, L.-C. N. Tokle and E. F. Brekke, ‘Belief propagation for marginal probabilities in multiple hypothesis tracking,’ To be published in Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)

Belief propagation for marginal probabilities in multiple hypothesis tracking

Odin Aleksander Severinsen, Lars-Christian Ness Tokle, Edmund Førland Brekke
Department of Engineering Cybernetics
The Norwegian University of Science and Technology (NTNU)
Trondheim, Norway

***Abstract*— This paper explores evaluation of association marginals in multiple hypothesis tracking. The work builds upon recent results where loop belief propagation (LBP) has been used in single-hypothesis cases. There are two contributions in the paper. The first is a novel factor graph representation of the joint multihypothesis association posterior. The second contribution is two algorithms that both use LBP to evaluate association marginals. The first method uses total probability in conjunction with hypothesis-conditioned LBP, and is called PHD-LBP. The second method is an LBP algorithm running directly on the full multihypothesis association graph with novel, specialized message definitions that are derived in this paper 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.**

I. Introduction

In order to do data association in multitarget tracking, one needs to build what is called *association hypotheses*. As time progresses, the number of possible hypotheses that can be made from the measurements that one receives grows exponentially, and so introducing assumptions and approximations are required for performing data association online. A typical approach in the target tracking literature is to model the *association posterior* as a *factor graph*, which encodes the underlying structure in the data association problem as a graph. The desired *association marginals* can then be approximated with the *loopy belief propagation* (LBP) algorithm, which takes considerably less computations to do compared to brute-force hypothesis enumeration for acceptable loss in accuracy.

A common denominator in all of these applications of factor graphs for data association in target tracking is that they are used in a *single-hypothesis* setting, where the number of association hypotheses after each timestep is approximated by a single hypothesis. The novelty in the present work is to propose a factor graph representation of the *joint multihypothesis association posterior* that appears in multiple hypothesis tracking, where we allow multiple association hypotheses after each timestep. We also present two different approaches that generalize the efficient LBP scheme presented in [1] to a multihypothesis setting.

This work was supported in part by the Research Council of Norway through Projects 295033 and 333917.

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 with a fraction of the computations needed for an exact solution. The second reason is for *track management* in the multitarget tracking filter *Poisson multi-Bernoulli mixture* (PMBM) that was first presented in [2] 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 [3] 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 [1] for multihypothesis scenarios we pave the way for achieving the same in a multihypothesis scenario.

The paper is structured as follows. First, related work on use of LBP in target tracking is presented in Section II. Then, required background theory and model assumptions are presented in Section III before the novel factor graph representation and association methods are presented in Section IV. The results from testing the methods on a simulated dataset are presented in Section V. The paper is then concluded in Section VI.

II. Related work

The present work uses factor graphs, a probabilistic graph representation first presented by Kschischang et. al in [4], to efficiently approximate a solution to the data association problem. They also describe how to conduct *belief propagation* on such a graph, an inference algorithm first presented by Pearl [5]. Early work on use of factor graphs in target tracking was done by Chen, Cetin et. al in [6]–[9] 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 [1], [10] augments the data association problem by overparameterizing of association variables which allows for formulating a bipartite matching graph and applies LBP to efficiently and quickly compute approximate association marginals that can be used in a MTT filter, such as in [2]. Together with Vontobel in [11]

they prove that this graphical representation exhibits certain properties that guarantees convergence of LBP, a particularly desirable property. In one of their latest work [12] they do approximate marginalization on an association graph similar to the one in [1] generalized for single-hypothesis inference jointly over multiple measurement scans. 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 [13] the authors embed the data association method presented in [1] 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 [14] and the presence of an unknown number of targets [15].

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 [16]. In [17] the same authors use BP to perform data fusion of radar and AIS (Automatic Identification System) data.

III. Preliminaries

A. Factor graphs

A factor graph 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

$$f(x_{\mathcal{V}}) = \prod_{a} f_a(x_{\mathsf{N}(a)}) \tag{1}$$

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.

B. Belief propagation

Belief propagation is an algorithm for doing efficient inference on probabilistic graphs with *tree structure* by exploiting the structure of the graph. For loopy graphs, we define the iterative computation of *messages* between nodes, given by

$$\mu_{a\to i}(x_i) \leftarrow \sum_{x_{\mathsf{N}(a)\setminus\{i\}}} f_a(x_{\mathsf{N}(a)}) \prod_{j\in\mathsf{N}(a)\setminus\{i\}} \mu_{j\to a}(x_j), \tag{2}$$

$$\mu_{i\to a}(x_i) \leftarrow \prod_{b\in\mathsf{N}(i)\setminus\{a\}} \mu_{b\to i}(x_i) \tag{3}$$

where $\mu_{a\to i}$ denotes the message from factor a to variable i and vice-versa for $\mu_{i\to a}$. The iterations are repeated until the messages converge or until some preset max number of iterations are done, and the messages are initialized to unity.

Perhaps the largest contribution to understanding the behavior of LBP came with the seminal work by Yedidia et. al [18] that demonstrated the properties of LBP and its connection to *variational inference*. Here, they describe the LBP algorithm as a constrained optimization of the *variational free energy* given a *trial density* q. The trial distribution is constructed to take the form of a simpler function that is feasible to do inference on. The particular constraint chosen is that q must factorize into the same nodes and edges as the original graph in a way that is exact for trees, and is called the *Bethe approximation* [19].

C. Target tracking models for data association

For the sake of simplicity we will make *linear Kalman filter* assumptions and the resulting tools for data association will be summarized below in a multitarget tracking framework. The reader is referred to [20] 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 is linear and Gaussian, such that we have

$$p(\boldsymbol{x}_k^t|\boldsymbol{x}_{k-1}^t) = \mathcal{N}\left(\mathbf{F}\boldsymbol{x}_{k-1}^t, \mathbf{Q}\right). \tag{4}$$

for an appropriate transition matrix $\mathbf{F}$ and covariance matrix $\mathbf{Q}$. To define the measurement model, assume we have k consecutive sets of measurements denoted by $Z_1 = \{\boldsymbol{z}_1^1, \ldots, \boldsymbol{z}_1^{m_1}\}, \ldots, Z_k = \{\boldsymbol{z}_k^1, \ldots, \boldsymbol{z}_k^{m_k}\}$. We use $\boldsymbol{z}_k^j$, $j \in \{1, \ldots, m_k\}$ to denote the jth measurement out of the m_k we receive in timestep k. The *measurement model* for a particular measurement $\boldsymbol{z}_k^j$ is $p(\boldsymbol{z}_k^j|\boldsymbol{x}_k^t)$ and is given by

$$p(\boldsymbol{z}_k^j|\boldsymbol{x}_k^t) = \mathcal{N}\left(\mathbf{H}\boldsymbol{x}_k^t, \mathbf{R}\right) \tag{5}$$

for an appropriate measurement matrix $\mathbf{H}$ and covariance matrix $\mathbf{R}$. Suppose that the *posterior distribution* $p(\boldsymbol{x}_{k-1}|Z_{1:k-1})$ of a track t in timestep $k-1$ is Gaussian. Together with the equations in (4) and (5), the *prior distribution* $p(\boldsymbol{x}_k^t|Z_{1:k-1})$ is given by

$$p(\boldsymbol{x}_k^t|Z_{1:k-1}) = \mathcal{N}\left(\hat{\boldsymbol{x}}_{k|k-1}^t, \mathbf{P}_{k|k-1}^t\right), \tag{6}$$

where $\hat{\boldsymbol{x}}_{k|k-1}^t$ and $\mathbf{P}_{k|k-1}^t$ denotes the predicted target state and covariance, respectively. The *likelihood distribution* is given by

$$p(\boldsymbol{z}_k|Z_{1:k-1}) = \mathcal{N}\left(\mathbf{H}\hat{\boldsymbol{x}}_{k|k-1}^t, \mathbf{H}\mathbf{P}_{k|k-1}^t\mathbf{H} + \mathbf{R}\right) \tag{7}$$

for each track t, and we will denote the value we get by evaluating the distribution for some track t in a particular measurement $\boldsymbol{z}_k^j$ by l^{tj}.

D. Track definition

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 and can be represented as a vector

$$\mathcal{I}^t = [i_1, \ldots, i_k] \tag{8}$$

where $i_l = \{0, \ldots, m_l, N\}$ for each $l \in \{1, \ldots, k\}$. The *nonexistence index* $i_l = N$ is used to indicate a track that has not been detected yet in timestep l, and as such "does

not exist". Thus, for a track initialized in timestep $L+1$ we must have that $i_l = N$ for $l = 1, \dots, L$.

E. Hypothesis definition

In this paper, we refer to an association hypothesis as a *tree*. 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 which will be denoted by the set a_k containing each association event a^t_k for each track t. The valid association events for each track is either $a^t_k = 0$, denoting misdetection, or $a^t_k = j$, $j = 1, \dots, m_k$ for m_k measurements denoting that measurement j is a detection of track t. An unassociated measurement j' is declared a new target, denoted by $j' \in \{n_k+1, \dots, n_k+m_k\}$ for n_k tracks.

1) 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 $r \in \{1, \dots, R\}$. 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

$$\theta^r_{1:k} = a_k \cup \theta^l_{1:k-1} \tag{9}$$

with base case $\theta^1_0 = \{\ \}$. Due to the definition in (8) we can make the definition in (9) more compact by referring to a hypothesis as a subset of all n_k track indices

$$\theta^r_{1:k} \subseteq \{1, \dots, n_k\} \tag{10}$$

where instead each scalar index $t \in \theta^r_{1:k}$ points to a vector $\mathcal{I}^t$.

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 (10) 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 do not exist in the hypothesis θ.

F. Joint association posterior

We assume the association posterior is given by

$$\mathsf{Pr}\{\theta^r_{1:k}|Z_{1:k}\} \propto \mathsf{Pr}\{\theta^l_{1:k-1}|Z_{1:k-1}\} \prod_{t:a^t=0} (1 - r^t_k P_d) \prod_{t:a^t>0} \frac{r^t_k P_d l^{ta^t}}{P_d \nu_k + \lambda} \tag{11}$$

where P_d is some constant detection probability, r^t_k denotes the *existence probability* of track t in timestep k, λ is the *clutter intensity*, ν_k denotes the arrival intensity of new targets in all of the valid target space, $\prod_{t:a^t=0}$ denotes the product over undetected tracks in timestep k under the hypothesis $\theta^r_{1:k}$, $a^t = j$ denotes the index of the measurement track t is associated with and $\prod_{t:a^t>0}$ denotes the product over all tracks t that are detected. A full derivation of (11) can be found in e.g. [2] and is omitted here.

IV. Multihypothesis data association

We will here present a novel factor graph representation of the joint multihypothesis association posterior, which is based upon the work in [1], but introduces two novelties. Firstly, we introduce the *hypothesis variable* θ, which extends the inference capabilities of the factor graph to be multihypothesis. Secondly, we introduce the *nonexistence state* $a^t = N$ to the association posterior 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.

Based on this, the new factorization can be derived as follows. We use the same overparameterization of track-measurement associations as in [1]. The track association variable a^t_k, defined in Section III-E, denotes the association of track t in timestep k. We additionally introduce the *measurement association variable* b^j_k, $j = 1, \dots, m_k$ with m_k being the number of measurements, defined as $b^j_k = t$ if measurement j is associated with track t and $b^j_k = 0$ if measurement j is a false alarm. We then require the compatibility factors γ^{tj} between the tracks a^t_k and measurements b^j_k which are given as

$$\gamma^{tj}(a^t_k, b^j_k) = \begin{cases} 0, & a^t_k = j \wedge b^j_k \neq t \ \vee \ a^t_k \neq j \wedge b^j_k = t \\ 1, & \text{otherwise} \end{cases} \tag{12}$$

in order to assign 0 probability to invalid association hypotheses that disobeys the standard at-most-one assumption. Given a prior hypothesis $\theta^l_{1:k-1}$, the distribution over the association hypothesis a_k then takes the form

$$\mathsf{Pr}\{a_k \mid \theta^l_{1:k-1}, Z_{1:k}\} \propto \prod_{t=1}^{n_k} \left(\psi^t(a^t) \prod_{j=1}^{m_k} \gamma^{tj}(a^t, b^j) \right) \tag{13}$$

where

$$\psi^t(a^t = 0) = 1 - r^t_k P_d, \tag{14}$$

$$\psi^t(a^t = j) = \frac{r^t_k P_d l^{tj}}{P_d \nu_k + \lambda}, \quad j \in \{1, \dots, m_k\} \tag{15}$$

which follows from (11).

We now introduce the required factors for the multihypothesis case. First, we add the prior factor $\varphi(\theta_{1:k-1})$,

$$\varphi(\theta_{1:k-1}) = \mathsf{Pr}\{\theta_{1:k-1} \mid Z_{1:k-1}\}, \tag{16}$$

for the prior hypothesis variable $\theta_{1:k-1}$ to the factorization, where we drop the superscript l as it can be any parent hypothesis. Additionally, we introduce the compatibility factor ζ^t between tracks a^t_k and $\theta_{1:k-1}$, which is defined by

$$\zeta^t(\theta_{1:k-1}, a^t_k) = \begin{cases} 1, & \begin{array}{l} t \in \theta_{1:k-1} \wedge a^t_k \neq N \\ \vee t \notin \theta_{1:k-1} \wedge a^t_k = N \end{array} \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

which we require to encode the nonexistence state $a^t_k = N$. The logical statement for $\zeta^t(\theta_{1:k-1}, a^t_k) = 1$ can be interpreted as one of two mutually exclusive requirements

that must be fulfilled. One of the requirements, the *existence consistency requirement*, is that $a_k^t = j$, $j = 0, 1, \ldots, m_k$, i.e. a track t can only be associated with misdetection or detection in the cases where $\theta_{1:k-1}$ takes the value of a prior hypothesis containing track t. The alternative requirement, the *nonexistence consistency requirement*, is that $a_k^t = N$, i.e. track t does not exist, only in the cases when $\theta_{1:k-1}$ takes the value of a hypothesis that does not contain track t. Again, the purpose of this factor is to assign 0 probability to invalid association hypotheses. We will now rewrite the expression in (11) in a factorized form that we can use to build a factor graph. Note that we have in (11) evaluated the hypothesis posterior in a specific child hypothesis $\theta_{1:k}^r$ which branches of the parent hypothesis $\theta_{1:k-1}^l$. To generalize the expression for all hypotheses $\theta_{1:k-1}$ we include the compatibility factor ζ^t from (17) and use $\psi^t(a_k^t = N) = 1$ for all $t \notin \theta_{1:k-1}$. This lets us arrive at the expression

$$\mathsf{Pr}\{\theta_{1:k}|Z_{1:k}\} \propto \varphi(\theta_{1:k-1}) \times \prod_{t=1}^{n_k}\left[\zeta^t(\theta_{1:k-1}, a_k^t)\psi^t(a_k^t)\prod_{j=1}^{m_k}\gamma^{tj}(a_k^t, b_k^j)\right]. \tag{18}$$

where n_k denotes the number of tracks.

An illustrative example of how such a factor graph can look like can be found in Figure 1 for a tracking scenario where we have three tracks a_k^1, a_k^2 and a_k^3 and one measurement b_k^1.

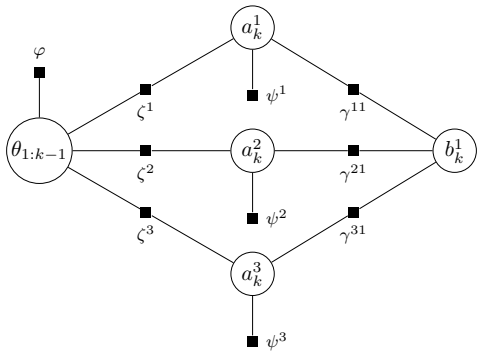


Fig. 1: *A toy example with three tracks a^1, a^2 and a^3 and one measurement b^1.*

A. Hypothesis-conditioned loopy belief propagation

Before presenting the main result of this paper, 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 [1], for then to sum these together with appropriate scaling. By total probability and Bayes' rule, the marginal can be written as

$$\mathsf{Pr}\{a_k^t \mid Z_{1:k}\} \propto \sum_{\theta_{1:k-1}} \Big\{\mathsf{Pr}\{a_k^t \mid \theta_{1:k-1}, Z_{1:k}\} \times p(Z_k \mid \theta_{1:k-1}, Z_{1:k-1})\varphi(\theta_{1:k-1})\Big\} \tag{19}$$

For tracks that exist in the prior hypothesis $\theta_{1:k-1}$, the marginal $\mathsf{Pr}\{a_k^t \mid \theta_{1:k-1}, Z_{1:k}\}$ can be computed with LBP, setting $\mathsf{Pr}\{a_k^t = N \mid \theta_{1:k-1}, Z_{1:k}\} = 0$. For tracks that does not exist in the prior hypothesis we set $\mathsf{Pr}\{a_k^t = N \mid \theta_{1:k-1}, Z_{1:k}\} = 1$ and all other association events to 0. What remains is to compute the hypothesis-conditioned set likelihood $p(Z_k \mid \theta_{1:k-1}, Z_{1:k-1})$. Computing it exactly involves full hypothesis enumeration, which is in general infeasible. Instead, we use approximations from the PHD filter [21] which is based on random finite set theory (RFS). Going forward we will refer to this method by the name PHD-LBP. For the sake of brevity, the derivation details are left out. The result is that we can approximate the hypothesis-conditioned likelihood with

$$p(Z_k|\theta_{1:k-1}, Z_{1:k-1}) \approx K\exp\left(-\sum_{t=1}^{n_k} r_k^t P_d\right) \times \prod_{j=1}^{m_k}\left[\left(\sum_{t=1}^{n_k}\frac{r_k^t P_d l^{tj}}{P_d\nu_k + \lambda}\right) + 1\right] \tag{20}$$

where K is a common constant for all $\theta_{1:k-1}$ and is cancelled after the final normalization. This approximation approximates the true Binomial set distribution by a Poisson set distribution with the same PHD.

B. Multihypothesis loopy belief propagation

The following section will present the main result of this paper. For doing LBP on the full multihypothesis factor graph, we define the track-to-measurement message $\mu_{t\to j}$, the measurement-to-track message $\nu_{t\to j}$, the hypothesis-to-track message σ_t and the track-to-hypothesis message ρ_t. We only use the subscript t for σ_t and ρ_t as all these messages unambiguously are either from or to the prior hypothesis variable $\theta_{1:k-1}$, respectively. The message directions are illustrated in Figure 2. In principle, doing LBP is matter

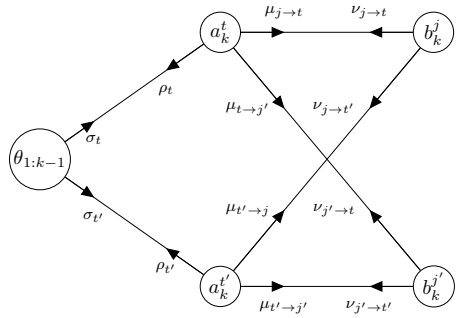


Fig. 2: *Simplified illustration of message directions in association graph.*

of computing the messages in (2) and (3). By inserting the

factors from (18) into (2) and (3), the general LBP equations take the form

$$\mu_{t\to j}(b_k^j) = \sum_{a_k^t} \psi^t(a_k^t)\gamma^{tj}(a_k^t, b_k^j)\left(\prod_{j'\neq j}\nu_{j'\to t}(a_k^t)\right)\sigma_t(a_k^t), \tag{21}$$

$$\nu_{j\to t}(a_k^t) = \sum_{b_k^j}\gamma^{tj}(a_k^t, b_k^j)\prod_{t'\neq t}\mu_{t'\to j}(b_k^j), \tag{22}$$

$$\sigma_t(a_k^t) = \sum_{\theta_{1:k-1}}\zeta^t(\theta_{1:k-1}, a_k^t)\varphi(\theta_{1:k-1})\prod_{t'\neq t}\rho_{t'}(\theta_{1:k-1}), \tag{23}$$

$$\rho_t(\theta_{1:k-1}) = \sum_{a_k^t}\zeta^t(\theta_{1:k-1}, a_k^t)\psi_t(a_k^t)\prod_j \nu_j(a_k^t), \tag{24}$$

where $\sum_{b_k^j}$ denotes the sum over all values $b_k^j \in \{0, 1, \ldots, n_k\}$, $\sum_{a_k^t}$ denotes the sum over all values $a_k^t \in \{0, 1, \ldots, m_k, N\}$, $\sum_{\theta_{1:k-1}}$ denotes the sum over all values $\theta_{1:k-1} \in \{\theta_{1:k-1}^1, \ldots, \theta_{1:k-1}^L\}$ 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 [1], 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_k^t, b_k^t and $\theta_{1:k-1}$, *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.

Due to the compatibility factors ζ^t and γ^{tj}, the message values are reduced to the distinct values

$$\mu_{t\to j}(b_k^j) = \begin{cases}\psi^t(j)\left(\prod_{j'\neq j}\nu_{j'\to t}(j)\right)\sigma_t(j), & b_k^j = t\\ \sum_{a_k^t\neq j}\psi^t(a_k^t)\prod_{j'\neq j}\nu_{j'\to t}(a_k^t)\sigma_t(a_k^t), & b_k^j \neq t\end{cases} \tag{25}$$

$$\nu_{j\to t}(a_k^t) = \begin{cases}\prod_{t'\neq t}\mu_{t'\to j}(t), & a_k^t = j\\ \sum_{b_k^t\neq t}\prod_{t'\neq t}\mu_{t'\to j}(b_k^j), & a_k^t \neq j\end{cases} \tag{26}$$

$$\sigma_t(a_k^t) = \begin{cases}\sum_{\theta:\, t\in\theta}\varphi(\theta_{1:k-1})\prod_{t'\neq t}\rho_{t'}(\theta_{1:k-1}), & a_k^t \neq N\\ \sum_{\theta:\, t\notin\theta}\varphi(\theta_{1:k-1})\prod_{t'\neq t}\rho_{t'}(\theta_{1:k-1}), & a_k^t = N\end{cases} \tag{27}$$

$$\rho_t(\theta_{1:k-1}) = \begin{cases}\sum_{a_k^t\neq N}\psi_t(a_k^t)\prod_j \nu_j(a_k^t), & t\in\theta_{1:k-1}\\ \psi_t(N)\prod_j \nu_j(N). & t\notin\theta_{1:k-1}\end{cases} \tag{28}$$

where $\sum_{\theta:\, t\in\theta}$ and $\sum_{\theta:\, t\notin\theta}$ denotes the sum over all prior hypotheses $\theta_{1:k-1}$ containing and not containing the track t, respectively, and $\sum_{a_k^t\neq j}$ and $\sum_{b_k^t\neq t}$ denotes the sum over all valid values of a_k^t and b_k^j except for j and t, respectively. We then normalize the messages appropriately with $\mu_{t\to j}(b_k^j \neq t)$, $\nu_{j\to t}(a_k^t \neq j)$, $\sigma_t(a_k^t \neq N)$ and $\rho_t(\theta_{1:k-1}), t\notin\theta_{1:k-1}$ to get the scalar message definitions

$$\mu_{t\to j} = \frac{\psi^t(j)}{\psi^t(0) + \sum_{j'\neq j, j'>0}\psi^t(j')\nu_{j'\to t} + \sigma_t}, \tag{29}$$

$$\nu_{j\to t} = \frac{1}{1 + \sum_{t'\neq t, t'>0}\mu_{t'\to j}}, \tag{30}$$

$$\sigma_t = \rho_t \cdot \frac{\sum_{\theta\,:\, t\notin\theta}\varphi(\theta)\prod_{t'\in\theta}\rho_{t'}}{\sum_{\theta\,:\, t\in\theta}\varphi(\theta)\prod_{t'\in\theta}\rho_{t'}}, \tag{31}$$

$$\rho_t = \psi^t(0) + \sum_{j=1}^{m_k}\psi^t(j)\nu_{j\to t}, \tag{32}$$

where $\prod_{t'\in\theta}$ denotes the product over all tracks $t \in \theta_{1:k-1}$.

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

$$\hat{p}(a_k^t|Z_{1:k}) \propto \begin{cases}\psi^t(0), & a_k^t = 0\\ \psi^t(j)\nu_{j\to t}, & a_k^t = 1, \ldots, m_k\\ \sigma_t, & a_k^t = N\end{cases} \tag{33}$$

while the measurement marginals are computed with

$$\hat{p}(b_k^j|Z_{1:k}) \propto \begin{cases}1, & b_k^j = 0,\\ \mu_{t\to j}, & b_k^j = 1, \ldots, n_k\end{cases} \tag{34}$$

and the prior hypothesis posterior

$$\hat{p}(\theta_{1:k-1}|Z_{1:k}) \propto \varphi(\theta_{1:k-1})\prod_{t\in\theta}\rho_t \tag{35}$$

V. Simulation results

The proposed methods for approximate marginals presented in Section IV were tested on a large, simulated dataset consisting of 1397 simulated radar scans in 2 dimensions. For each timestep, the methods are tested separately on each cluster of tracks. Extracting the cluster data from the timestep data showed that there are in total 111887 clusters in the dataset. The reader is referred to [22] for more details.

A. The methods compared

Three methods are compared in the following results. The two first methods are the methods MH-LBP and PHD-LBP, presented in Section IV-B and Section IV-A, respectively. Additionally, as a benchmark or best-case to compare with, a hypothesis-conditioned LBP similar to PHD-LBP was also tested that used the exact hypothesis-conditioned normalization constant instead of the PHD approximation. We here mean "exact" as the normalization constant to (13) which can be found by doing full posterior hypothesis enumeration for each prior hypothesis. The purpose of this was to isolate errors from the approximate normalization constant in order to better capture the properties and failure modes of LBP on the multihypothesis association problems in question.

B. Discussion

The plots in Figure 3 show the survival function of the empirical distribution of the marginal errors. Comparing the performance of MH-LBP with that of PHD-LBP we see that overall, the two approaches are similar. Notably, MH-LBP seems to perform better in particular for misdetections than PHD-LBP, 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 into the nonexistence probability from the remaining probabilities estimated from LBP. 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 in the same way as we have for nonexistence.

In Figure 4 the PHD approximation normalization constant is compared to the exact normalization constant in a correlation plot with logarithmic scale. 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.

The correlations between the exact marginals and MH-LBP, PHD-LBP and LBP with exact normalization constants, respectively, can be found in Figure 5.

In the correlation plot for MH-LBP most marginals are well correlated with the exact 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. 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 as MH-LBP underestimates the probabilities. To conclude, it can seem from the correlation

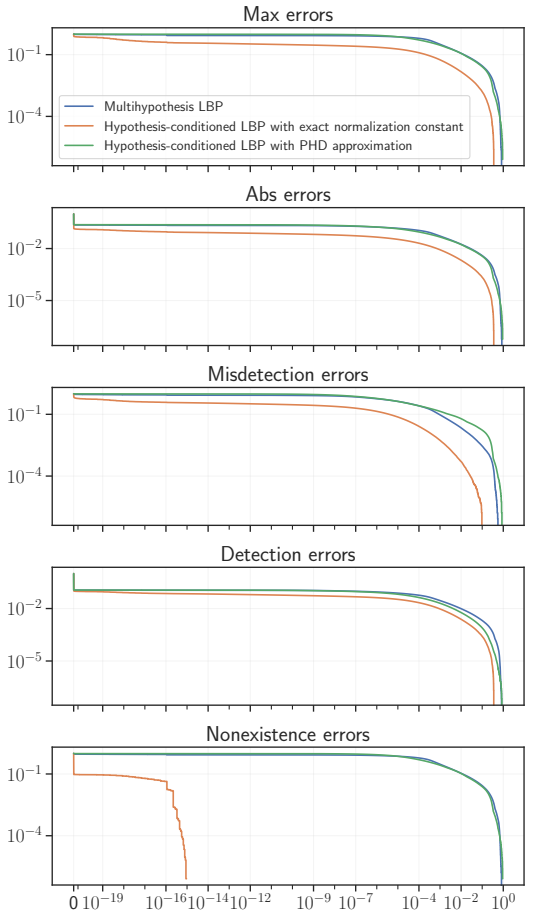


Fig. 3: *Survival function for different errors. "Max errors" means the maximum of the absolute value of the marginal error for each marginal distribution, while "Abs errors" stands for the absolute value for all marginal errors. Note that both the y-axis and x-axis are logarithmic. The vertical line at $x = 0$ is due to all the marginals that estimated with zero error. The legend is the same for all plots and is only included in the first plot for clarity.*

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.

The correlation plot for PHD-LBP 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 PHD-LBP 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 could, again, be the the flatness of the PHD approximation that amplifies the conservative behavior that we saw in MH-LBP.

Lastly, we will inspect the correlation plot for the best-case LBP with exact normalization constant. Overall, the

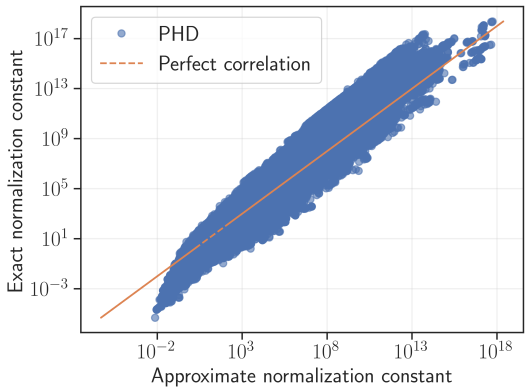


Fig. 4: *Correlation plot between estimated normalization constant and true normalization constant with logarithmic scaling.*

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 raises 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.

1) 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. They prove mathematically in [1] that the track-to-measurement and measurement-to-track messages, $\mu_{t\to j}$ and $\nu_{j\to t}$, respectively, must converge. 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 [12] they consider a similar association problem that is multiscan instead of multihypothesis 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. In [1], results show that the accuracy of LBP is tied to the SNR of the problem, where lower SNR seems to improve accuracy and 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 [23] 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

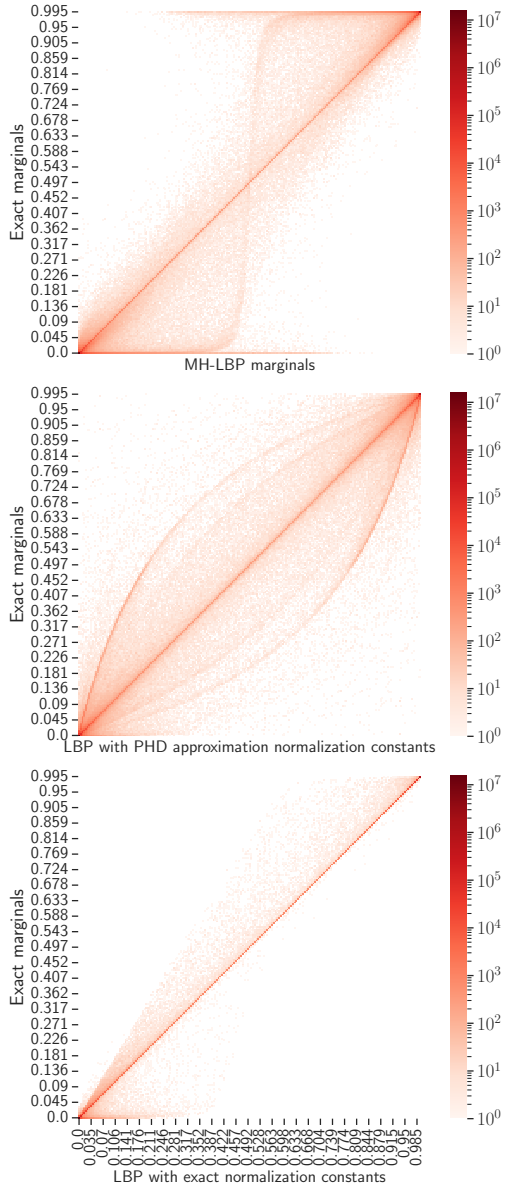


Fig. 5: *Heatmap showing correlation between the marginals of the methods compared and exact marginals. Note that the colors are logarithmic.*

misdetection probablities do appear in the priors of the tracks in our factor graph, it was tested with considerable higher misdetection probabilities. After adding 2.5 to the log misdetections we achieve convergence.

VI. Conclusion

This paper has proposed two methods, MH-LBP and PHD-LBP, for computing approximate association marginals in a multihypothesis tracker based on LBP and a novel factor graph representation of the multihypothesis association hypothesis posterior. The methods are tested on a simulated dataset and compared with a best-case comparison that does hypothesis-conditioned LBP in the same manner as PHD-LBP, but with exact hypothesis-conditioned likelihood. The results show that both MH-LBP and PHD-LBP perform well in most cases. The largest differences between was attributed to the Poisson approximation of the hypothesis-conditioned likelihood. 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.

References

[1] 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.

[2] 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.

[3] J. L. Williams, "Graphical model approximations of random finite set filters," *CoRR*, 2011. [Online]. Available: `http://arxiv.org/abs/1105.3298`.

[4] 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.

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

[6] 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.

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

[8] L. Chen, M. J. Wainwright, M. Cetin, 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.

[9] 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.

[10] 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.

[11] P. O. Vontobel, "The bethe permanent of a non-negative matrix," *IEEE Transactions on Information Theory*, vol. 59, no. 3, 1866–1901, 2013.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] H. A. Bethe, "Statistical theory of superlattices," *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, vol. 150, no. 871, 552–575, 1935.

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

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

[22] E. F. Brekke and A. G. Hem, "A long simulation scenario for evaluation of multi-target tracking methods," *Accepted for publication at ICECCME*, 2023.

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

Bibliography

[1] O. A. Severinsen, L.-C. N. Tokle and E. F. Brekke, 'Belief propagation for marginal probabilities in multiple hypothesis tracking,' *To be published in Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)*,

[2] 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.

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

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

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

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

[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] 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).

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

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] '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.

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

[26] 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.

[27] P. A. Regalia and J. M. Walsh, 'Optimality and duality of the turbo decoder,' *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1362–1377, 2007. DOI: 10.1109/JPROC.2007.896495.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT, USA: YBS Publishing, 1995.

[35] 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.

[36] 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.

[37] Á. F. García-Fernández, J. L. Williams, K. Granström and L. Svensson, 'Poisson multi-bernoulli mixture filter: Direct derivation and implementation,' *CoRR*, vol. abs/1703.04264, 2017. arXiv: 1703.04264. [Online]. Available: http://arxiv.org/abs/1703.04264.

[38] 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.

[39] 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).

[40] 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.

[41] 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.

[42] 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.

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

[44] 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.

[45] A. H. Land and A. G. Doig, 'An automatic method of solving discrete programming problems,' *Econometrica*, vol. 28, no. 3, p. 497, 1960. DOI: 10.2307/1910129.

[46] 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.

[47] A. G. Hem, 'Maritime multi-target tracking with radar and asynchronous transponder measurements,' M.S. thesis, NTNU, 2021. [Online]. Available: https://hdl.handle.net/11250/2781021.

[48] C. E. Bonferroni, 'Teoria statistica delle classi e calcolo delle probabilità,' Italian, *Pubbl. D. R. Ist. Super. Di Sci. Econom. E Commerciali di Firenze*, vol. 8, pp. 1–62, 1936.

[49] E. F. Brekke and A. G. Hem, 'A long simulation scenario for evaluation of multi-target tracking methods,' *Accepted for publication at ICECCME*, 2023.

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

[51] G. W. et. al, 'Lecture notes in 6.438 Algorithms for Inference,' *Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science*, 2021.

[52] B. Huang and T. Jebara, 'Approximating the permanent with belief propagation,' *CoRR*, vol. abs/0908.1769, 2009. arXiv: 0908.1769. [Online]. Available: http://arxiv.org/abs/0908.1769.