A GRAPHICAL MODEL REPRESENTATION OF THE TRACK-ORIENTED MULTIPLE HYPOTHESIS TRACKER

Andrew Frank, Padhraic Smyth, and Alexander Ihler

Department of Computer Science
University of California, Irvine

ABSTRACT

The track-oriented multiple hypothesis tracker is currently the preferred method for tracking multiple targets in clutter with medium to high computational resources. This method maintains a structured representation of the track posterior distribution, which it repeatedly extends and optimizes over. This representation of the posterior admits probabilistic inference tasks beyond MAP estimation that have yet to be explored. To this end we formulate the posterior as a graphical model and show that belief propagation can be used to approximate the track marginals. These approximate marginals enable an online parameter estimation scheme that improves tracker performance in the presence of parameter misspecification.

Index Terms— multi-target tracking, multiple hypothesis tracker, track-oriented, graphical model, parameter estimation

1. INTRODUCTION

Multitarget tracking is an important problem with applications spanning national defense, robotics, and consumer electronics. The general case is NP-hard, so a number of algorithms have been proposed over the years, each making a different set of approximations to achieve tractability. The currently favored algorithm for tracking in difficult environments with high computational resources is the track-oriented multiple hypothesis tracker (TOMHT) [1]. The TOMHT operates by maintaining a pruned set of potential tracks using a data structure called a track tree. Each subset of these tracks corresponds to a different data association hypothesis (explanation of the observed data), and at each time step the tracker solves a constrained optimization problem to pick out the hypothesis that best explains the data seen thus far.

Track trees and the constraints between them define the posterior probability distribution over tracks, and the aforementioned optimization problem amounts to maximum a posteriori (MAP) estimation. While trackers based solely on MAP estimation have proven effective in a number of real-world deployments, e.g. [2], they do not fully exploit

the TOMHT's representation of the posterior distribution. Specifically, the structured representation of the posterior admits efficient algorithms for estimating quantities such as track marginal probabilities and the partition function, which may be used to improve tracker performance.

The primary contributions of this paper are:

1. A formulation of the TOMHT's track posterior distribution as a factor graph.

2. Experimental results demonstrating that approximate track marginals derived from this factor graph enable online parameter estimation and can improve tracker performance.

2. PROBLEM FORMULATION

Let $\mathcal{Z} \subset \mathbb{R}^{d_z}$ denote the region of surveillance and $\mathcal{X} \subset \mathbb{R}^{d_x}$ the target state space. Targets are modeled as points $x \in \mathcal{X}$, evolving in discrete time according to a dynamics function $f_d(x^{k+1} \mid x^k)$. At each time step a target is detected with probability p_D, yielding an observation $z \in \mathcal{Z}$ distributed according to the observation model $f_o(z \mid x)$. Observations generated at a given time step k are grouped together into a set called a *scan*, denoted $\mathbf{z}^k$, and passed as input to the tracking algorithm. At each time step, the goal of the tracking algorithm is to return a set of trajectories, each representing a unique target and its motion through $\mathcal{X}$, that best explains the scans received thus far.

Birth and death events for tracks correspond to targets entering or leaving the surveillance region and may occur at any time. The number of track births at a given time step is modeled as a Poisson random variable with rate λ_ν, and initial target states are distributed uniformly over $\mathcal{X}$. Existing tracks may die with probability p_γ.

Each scan may also include observations not generated from any target (false alarms). The number of false alarms in each scan is modeled as being Poisson with rate λ_ϕ, and they are assumed to be uniformly distributed throughout $\mathcal{Z}$.

We assume the availability of an efficient single-target tracking subroutine. This permits the common reduction of multitarget tracking to the data association problem: if we
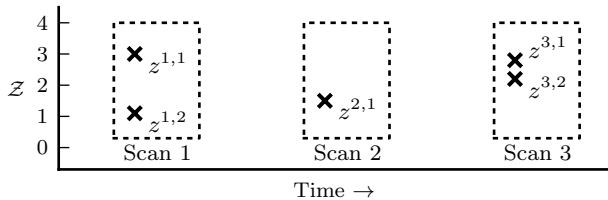
Fig. 1. A small example problem three scans of data.

Fig. 2. Track trees generated from the observations in Figure 1. To save space, trees created after scan 1 are omitted.

can split $z^{1:k}$ into groups, each corresponding to a single target, then we can delegate state estimation responsibilities to the single-target tracking module.

Figure 1 illustrates a small example that will serve to introduce some terminology. The notation $z^{k,j}$ represents the jth observation of scan k. Note that while the first index (scan number) provides a meaningful temporal ordering of the observations, the second (within-scan index) is arbitrary – it does not contain any information regarding the identity of the target that generated it.

Define a *track* to be a set of observation indices, e.g. $\{(1,2),(2,1),(3,2)\}$ in the example in Figure 1. A *hypothesis* is a set of tracks, each representing a unique target. If an observation is not included in any of the tracks in a hypothesis, it is assumed to be a false alarm. We represent hypotheses with indicator vectors $\boldsymbol{\tau}$ in which each element corresponds to a possible track. If the ith element $\boldsymbol{\tau}_i$ is 1, the hypothesis includes the corresponding track. A hypothesis represents a complete explanation of the observed data. Thus, data association can be formulated as an optimization problem where the goal is to find the most likely hypothesis:

$$\boldsymbol{\tau}^* = \arg\max_{\boldsymbol{\tau}} \Pr(\boldsymbol{\tau} \mid z^{1:k}) \qquad (1)$$

It is commonly assumed that each target can only give rise to one observation per scan and that each observation is the result of at most one target. Hypotheses that violate these assumptions are assigned zero prior probability.

3. BACKGROUND AND PRIOR ART

3.1. Track-oriented MHT

The TOMHT was introduced as an alternative to its closely-related predecessor, the hypothesis-oriented MHT (HOMHT) [3]. Both approaches take a deferred-decision approach in which they maintain the complete set of possible data associations within a sliding window, putting off hard decisions as long as possible. The TOMHT is unique in its representation of potential hypotheses: it uses a data structure called the *track tree* to represent hypotheses implicitly, as opposed to the HOMHT's explicit enumeration. This allows the TOMHT to represent a much larger set of hypotheses.

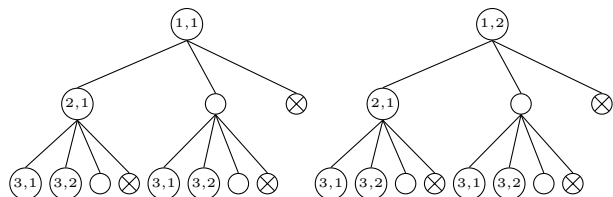A track tree is simply a rooted tree in which each node corresponds to an observation and every root-leaf path represents a possible track. As new scans arrive, existing track trees are extended to include the new observations and new trees are created to represent possible new targets. Pseudo-observations are included in each scan to represent missed detections and track deaths. Figure 2 illustrates two of the track trees resulting from the example data shown in Figure 1.

Kurien [1] showed that the log probability of a hypothesis could be written, up to a constant, as a sum involving one term for each constituent track:

$$\log \Pr(\boldsymbol{\tau} \mid z^{1:k}) = C + \sum_{i=1}^{|\mathcal{T}|} \boldsymbol{\tau}_i s_i \qquad (2)$$

The terms s_i, called *track scores*, lend themselves to efficient recursive computation. Furthermore, their values specify the optimization problem posed in (1).

3.2. Factor graphs

A factor graph [4] $\mathcal{G}$ is defined by a variable set $\mathcal{V}$ and factor set $\mathcal{F}$. Each factor f_i is a non-negative, real-valued function defined over a subset of the variables $\mathcal{V}_i$. Together the factors define a joint probability distribution: $\Pr(\mathcal{V}) = \frac{1}{Z} \prod_i f_i(\mathcal{V}_i)$, where $Z = \sum_{\mathcal{V}} \prod_i f_i(\mathcal{V}_i)$ is a normalizing constant called the partition function.

Numerous general-purpose approximate algorithms exist for estimating the MAP, marginals, and partition function of distributions written in this form. In this work we make use of a classic algorithm for approximate marginalization called loopy belief propagation (BP) [4]. While BP does not offer general guarantees regarding the accuracy of its results, it has been used with great success in many applications from error-correcting codes [5] to computer vision [6].

4. TRACK TREES AS FACTOR GRAPHS

In Section 2 we introduced two basic assumptions: each target produces at most one observation per scan, and each observation is the result of at most one target. We now formulate a factor graph that explicitly encodes these assumptions as constraints between random variables.

The *track tree factor graph* $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ contains one binary variable corresponding to each track tree node. Each $v_i \in \mathcal{V}$ serves as an indicator variable for the *partial* track
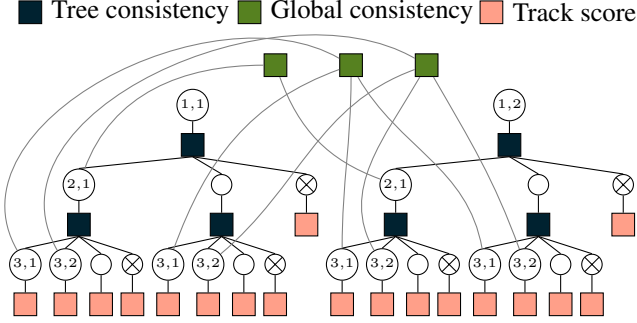
Fig. 3. The track tree factor graph corresponding to the track trees in Figure 2.

terminating in its corresponding node in the track tree. If a variable v_i corresponds to a track tree leaf node, we call it a *track indicator variable*. The set of such variables is denoted by $\mathcal{T}$. Every assignment to $\mathcal{T}$ uniquely specifies a hypothesis, but many such hypotheses violate our assumptions. The set $\mathcal{F}$ contains a collection of factors that encode our assumptions and assign probability zero to all invalid hypotheses.

We split these constraint factors into two groups: *tree consistency* factors and *global consistency* factors. Tree consistency factors ensure that, for each subgraph of $\mathcal{G}$ representing a track tree, only the all-zero configuration and configurations including a single root-leaf path of ones have non-zero probability. To achieve this we create one factor, f_i^t, for each non-leaf variable v_i. Denote by $\boldsymbol{v}_{ch(i)}$ the variables that are children of v_i (borrowing the parent-child relationships from the corresponding track tree). Then we define f_i^t as follows:

$$f_i^t(v_i, \boldsymbol{v}_{ch(i)}) = \begin{cases} 1 & : & (v_i = 0 \text{ and } \sum_{v_k \in v_{ch(i)}} v_k = 0) \\ & & \text{or } (v_i = 1 \text{ and } \sum_{v_k \in v_{ch(i)}} v_k = 1) \\ 0 & : & \text{otherwise.} \end{cases}$$

Global consistency factors assign zero probability to configurations that include the same observation in more than one selected track. Denote by $\boldsymbol{v}_{z^{i,j}}$ the set of variables corresponding to observation $z^{i,j}$. We add one global consistency factor for each unique observation (not including the pseudo-observations) and define them as follows:

$$f_i^g(\boldsymbol{v}_{z^{i,j}}) = \begin{cases} 1 & : & \sum_{v_k \in v_{z^{i,j}}} v_k \leq 1 \\ 0 & : & \text{otherwise.} \end{cases}$$

Finally, we introduce a collection of *score factors*, f_i^s, to appropriately weight the valid hypotheses. There is one score factor for each $v_i \in \mathcal{T}$, defined as follows:

$$f_i^s(v_i) = \begin{cases} e^{s_i} & : & v_i = 1 \\ 1 & : & v_i = 0, \end{cases}$$

where s_i is the score for track i as defined in equation 2.

Thus, the probability mass function represented by the factor graph may be written as

$$\Pr(\mathcal{V}) \propto \prod_{v_i \notin \mathcal{T}} f_i^t(v_i, \boldsymbol{v}_{ch(i)}) \prod_{z^{i,j}} f_{z^{i,j}}^g(\boldsymbol{v}_{z^{i,j}}) \prod_{v_i \in \mathcal{T}} f_i^s(v_i)$$

Figure 3 illustrates the factor graph resulting from the example input of Figure 1. An instantiation of the variables will evaluate to the exponentiated sum of the selected track scores if it corresponds to a valid hypothesis, and zero otherwise.

5. EM FOR ONLINE PARAMETER ESTIMATION

The parameters of the dynamics model can significantly affect tracking performance. It is often useful to be able to adjust the parameters as data is received, either because training data are unavailable or target behavior is expected to vary over time.

In single-target tracking, parameter estimation is often carried out using the EM algorithm, treating the target state as "missing data." The same strategy applies to multitarget tracking, with the caveat that measurement associations comprise additional missing data. The conditional expectations required for EM in this context take the following form:

$$\Pr(\boldsymbol{\tau}_i^k = 1 \mid \boldsymbol{z}^{1:k}) \int_{\boldsymbol{x}^{1:k}} g(\cdot) \Pr(\boldsymbol{x}^{1:k} \mid \boldsymbol{\tau}_i^k = 1, \boldsymbol{z}^{1:k}), \quad (3)$$

where $g(\cdot)$ is some function of the target state variable. The integral in this equation is the same conditional expectation that would be required in the single-target case. Thus, parameter estimation hinges on being able to compute the first term: the marginal probability of a track indicator variable.

In the HOMHT, the marginal probability of a track is computed as the sum of the probabilities of all hypotheses containing that track. In the TOMHT this computation is more involved: since hypotheses are not explicitly represented, it is not obvious how to efficiently sum over them. We propose to approximate the required marginals using belief propagation on the track tree factor graph. Recall that every track has a corresponding leaf variable in the factor graph. After running BP we can simply query the marginal of the appropriate leaf node for an estimate of the track marginal probability.

The marginals computed by BP are not exact, so this pseudo-EM algorithm will not monotonically increase the likelihood. However, our experiments suggest that the quality of the approximate marginals is sufficient for this application.

6. EXPERIMENTAL RESULTS

We conducted an empirical evaluation of the online EM algorithm on a set of simulated tracking problems, a typical example of which is illustrated in Figure 4. The data follow linear Gaussian dynamics and observation models:

$$f_d(x^{k+1} \mid x^k) = \mathcal{N}(Ax^k, Q) \qquad f_o(z \mid x) = \mathcal{N}(Hx, R)$$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad Q = \begin{bmatrix} .1^2 & 0 \\ 0 & .2^2 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad R = \begin{bmatrix} 0.25^2 \end{bmatrix}$$

In all trials we fixed the following parameters: $p_D = 0.95$, $\lambda_\phi = 1$, $\lambda_\nu = 1$, $p_\gamma = 0.1$. We also configured the tracker with the true values of A, H, and R, but experimented with
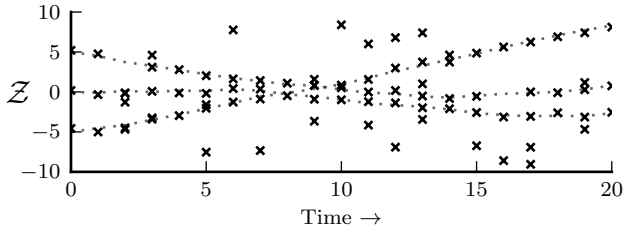
Fig. 4. A typical test problem used in our evaluation. Targets begin at locations -5, 0, and 5, with velocities 0.5, 0, and -0.5, respectively. True target paths are shown as dotted gray lines.
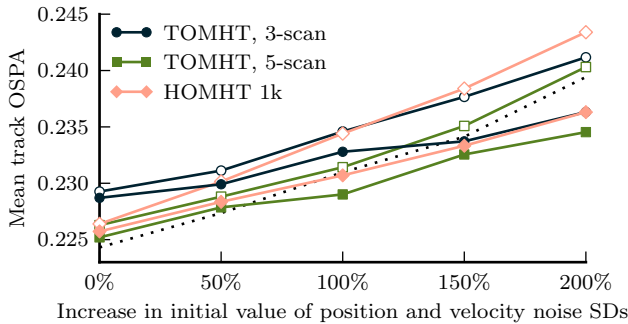

Fig. 5. Tracking performance as a function of parameter misspecification. Filled and empty markers indicate performance with and without EM, respectively. The dotted black line represents the best performance possible without online learning.

misspecifying Q to varying degrees, increasing the diagonal elements beyond their true values. This simulates the effect of the "cautious overestimation" one might engage in with hopes of avoiding the catastrophic effects of an underestimate. We consider four levels of misspecification, increasing the standard deviations by 50% of their original value each time.

To measure tracker performance we used the "OSPA for tracks" metric [7]. This metric accounts for localization error (distance between estimated and true target state), cardinality error (predicting the incorrect number of targets), and labeling error (swapping states among one or more tracks). Lower scores are better, and in our evaluation we parameterized the metric such that the worst possible score is 0.5.

Figure 5 plots the mean track OSPA score for three different tracker configurations over 100 simulated problems. TOMHT 3-scan and TOMHT 5-scan differ only in their degree of pruning, with TOMHT 5-scan being the more expensive variant. HOMHT 1k is a hypothesis-oriented MHT implemented for comparison purposes. The dotted black line corresponds to a HOMHT run without online estimation but with the capacity to maintain 100,000 hypotheses. In this example, that serves as a reasonable approximation of the best performance achievable without online estimation.

In all cases performance degrades as parameter misspecification increases, and in all cases online estimation is able to recover some of that lost performance. The magnitude of
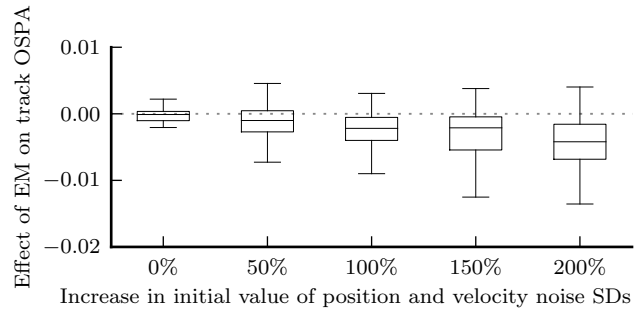

Fig. 6. The effect of online parameter estimation on tracking performance in each individual trial with the 5-scan TOMHT algorithm. Negative values indicate improved performance with online parameter estimation.

the improvement is comparable in the HOMHT and TOMHT cases, indicating that the approximate marginals from BP are not causing significant problems.

Figure 6 shows how the effect of EM varied among the 100 simulated problems. As expected, the benefit of parameter estimation increases with the degree of misspecification.

7. CONCLUSION

The track tree factor graph is a formalization of the TOMHT's posterior distribution over tracks in the language of graphical models. We demonstrated that this formulation enables approximate computation of track marginal probabilities, which may then be used to perform online parameter estimation. The factor graph representation also supports approximate partition function and mixed maximization / marginalization queries. These quantities could be useful in developing new methods for model selection and track pruning, which is a promising direction for future work.

References

[1] T. Kurien, "Issues in the design of practical multitarget tracking algorithms," *Multitarget-Multisensor Tracking: Advanced Applications*, 1990.

[2] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, 2004.

[3] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, 1979.

[4] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, 2001.

[5] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, 1998.

[6] J. Sun, N. Zheng, and H. Shum, "Stereo matching using belief propagation," *IEEE TPAMI*, vol. 25, 2003.

[7] B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo, "A metric for performance evaluation of multi-target tracking algorithms," *IEEE Trans. Signal Process.*, vol. 59, 2011.