XMAP: Track-to-Track Association with Metric, Feature, and Target-type Data

J. Ferry
Metron, Inc.
Reston, VA, U.S.A.
ferry@metsci.com

Abstract - The Extended Maximum A Posteriori Probability (XMAP) method for track-to-track association is based on a formal, Bayesian methodology for incorporating metric, feature, and target-type data. The metric component improves upon the classical derivation of the adaptive threshold to produce a more robust alternative, which can handle clusters with very few tracks and tracks with large covariances. The feature and target-type components are treated jointly, allowing for the possibility that the performance of the feature extractor depends on target type. This coupling allows feature information to be interpreted differently depending on the results of a target classifier—from a feature measurement being deemed accurate within within a small tolerance, to the measurement being thrown out altogether. A key innovation in the derivation is the non-informative noise assumption used in the feature measurement model, which gives a simple, robust form to the results.

Keywords: Data association, feature, target type, adaptive threshold, noise model.

1 Introduction

The XMAP (eXtended Maximum A Posteriori probability) method [1] is a principled, Bayesian technique for computing the optimal track-to-track association of data. It begins with simple, general assumptions about the nature of the probability distributions of (a) a set of targets about which one is attempting to gain information, and (b) the errors made in gathering this information. We will assume there is a number of “sensors” (often, but not necessarily, physical sensors), and that on each sensor there are a number of tracks, each of which provides information about some target. The XMAP method produces a formula for the probability of each possible association of the tracks on the various sensors. An association is a specification of which tracks refer to a common target. Getting the correct association is a necessary prerequisite to fusing the data from the sensors: incorrect association leads to the fusion of mismatched information. The association selected by the XMAP method may justly be called optimal: armed with the formula for the probability of each association, the XMAP method chooses the association with maximal probability.

XMAP is an extension of Mori and Chong’s MAP procedure [2]. The MAP method was formulated with only metric data in mind, and in the two-sensor case recovers the traditional solution to the association problem [3] (involving a cost matrix whose entries are chi-squared distances) except that it produces naturally a threshold for association. Other methods either force as many associations as possible, or have some ad hoc scheme for determining whether the cost of associating two tracks is sufficiently low to associate them.

The formulation of XMAP may be divided into two parts. The first is the derivation of the probability of an association, which is presented in Appendix A. This derivation is essentially equivalent to the derivation of MAP given in [4], but is simpler as it employs less mathematical machinery (such as random sets). The chief difference in the derivation of XMAP is that it works with an abstract space rather than a metric space. The second part of the derivation of XMAP is giving a useful structure to the abstract space it employs, and reasonable assumptions about the probabilities on that space. This latter part is the topic of this paper.

We make the following assumptions about the data. We assume that the targets are independently and identically distributed (i.i.d.) and that there are no split or merged tracks: i.e., each target produces at most one track on a given sensor, and each track on a given sensor arises from at most one target. We assume there are no false alarms. We assume that the error of each track depends only on the sensor and/or information carried by the track (such as a covariance matrix for a Gaussian distribution). See Bar-Shalom and Chen [5] for the extension to tracks with correlated errors. Finally, we assume that the prior on the number of targets is Poisson distributed with mean \( \nu \). See Mori and Chong [4] or Appendix A for the extension to non-Poisson distributions.

Under these conditions, the probability of any association is proportional to the product of factors \( R_{ij}(\mathbf{z}_s^1, \mathbf{z}_s^2) \), where the product runs over all pairs \((i, j)\) such that the association pairs track \( i \) on sensor 1 with track \( j \) on sensor 2. For \( s = 1, 2, \mathbf{z}_s^i \) is the measurement received on track \( i \) of sensor \( s \). We use “measurement” in a broad sense here, encompassing any kind of data sent to the association algorithm—in particular “measurement” could refer to the mean of a posterior
distribution obtained from the fusion of previous track information with a sensor measurement update. Each factor $R_{ij}(z^1_i, z^2_j)$ is given by the formula
\[
R_{ij}(z^1_i, z^2_j) = \frac{P_{ij}^{12}(z^1_i, z^2_j)}{\nu P^1_i(z^1_i) P^2_j(z^2_j)},
\]
(1)
where the probabilities $P^1_i(z^1_i)$ and $P_{ij}^{12}(z^1_i, z^2_j)$ are defined to be
\[
P^1_i(z^1_i) = \int P^1_D(x) Q^1_{ij}^{-1}(x) L^1_i(z^1_i|x) p^0(x) \, dx,
\]
(2)
and
\[
P_{ij}^{12}(z^1_i, z^2_j) = \int P^1_D(x) P^2_D(x) L^1_i(z^1_i|x) L^2_j(z^2_j|x) p^0(x) \, dx.
\]
(3)
These equations are derived in Appendix A, where they appear (in slightly different notation) as equations (61)–(63). In (2) and (3) $P^1_D(x)$ is the probability of sensor $s$ detecting a target in state $x$, whereas $Q^1_{ij}^{-1}(x) = 1 - P^2_D(x)$ is the probability of non-detection. The measurement likelihood function $L^1_i(z^1_i|x)$ is the probability density for a detected target in state $x$ produced in the measurement $x^1_i$ on sensor $s$. Finally, the prior probability density of a target being in state $x$ is denoted $p^0(x)$. Hence $P_{ij}^{12}(z^1_i, z^2_j)$ is the probability that track $i$ on sensor 1 and $j$ on sensor 2 arise from the same target, whereas $P^1_i(z^1_i)$ is the probability that track $i$ on sensor $s$ is not detected on the other sensor (sensor 3–$s$). Therefore $R_{ij}(z^1_i, z^2_j)$ is the probability ratio between the hypotheses that tracks $i$ and $j$ arise from the same versus different targets, the factor of $\nu$ being due to the hypothesis of different targets requiring one extra target in total. The MAP association is the one with maximal probability. Finding this association is a standard integer programming problem. It may be solved with the Jonker–Volgenant–Castañon (JVC) algorithm [6], for example.

The key to the XMAP procedure is in defining the state and measurement variables $x$ and $z^i_s$, and their probability structures. We begin this definition by expressing the state as $x = (x^M, x^f)$, where $x^M$ is the metric state and $x^f$ is the joint target-type/feature state. Similarly, the measurement will be assumed to consist of metric and joint target-type/feature components: $z^i_s = (z^{iM}_s, z^{if}_s)$. Assuming that the metric and target-type/feature prior and measurement likelihood functions are independent, then we may decompose the probability ratio $R_{ij}(z^1_i, z^2_j)$ as
\[
R_{ij}(z^1_i, z^2_j) = \nu^{-1} R^M_{ij}(z^{iM}_i, z^{jM}_j) R^f_{ij}(z^{if}_i, z^{jf}_j).
\]
(4)

Section 2 develops a formula for $R^M_{ij}(z^{iM}_i, z^{jM}_j)$, i.e., the metric component of XMAP. The result of this section is a refinement of the adaptive threshold of the MAP procedure, which is achieved by treating more carefully one of the standard assumptions that is made in the MAP derivation.

Section 3 develops a formula for $R^f_{ij}(z^{if}_i, z^{jf}_j)$, i.e., the joint target-type/feature component of XMAP. In this section we consider a very general scenario, in which a classifier is employed to ascertain target type, and the feature measurement likelihood function is allowed to depend strongly on this target type. Although XMAP works well when combining feature and classifier information under the assumption that they may be treated independently not only from the metric but from each other as well, it is often the case that the classification provides important information about the distribution of the underlying feature state for a given feature measurement. Section 3 presents the methodology that allows this information to be exploited in a mathematically correct manner.

## 2 Metric component

We write the metric component in (4) as
\[
R^M_{ij} = R_{ij}(z^{i1}_i, z^{j2}_j) = \frac{P^1_D(z^{i1}_i) P^2_D(z^{j2}_j)}{P^1_i(z^{i1}_i) P^2_j(z^{j2}_j)},
\]
(5)
where, dropping what would be a ubiquitous superscript “$M$,”
\[
P^i(z^{i1}_i) = \int L^i_s(z^{i1}_i|x) p^0(x) \, dx,
\]
(6)
\[
P^{12}_s(z^{i1}_i, z^{j2}_j) = \int L^1_s(z^{i1}_i|x) L^2_j(z^{j2}_j|x) p^0(x) \, dx.
\]
(7)

Here we have assumed that the detection probability is independent of the metric state: it will be incorporated into $R^f_{ij}$ in Section 3.

There are two standard assumptions made about metric information. First, the measurement likelihood functions are assumed to be Gaussian. Accordingly, we let $L^i_s(z^{i1}_i|x) = N(z^{i1}_i|x, V^i_s)$, i.e., a Gaussian with mean $x$ and covariance matrix $V^i_s$. Second, the prior distribution on the state is assumed to be $p^0(x) = I_S(x)/\Vol(S)$, where $S$ is some region of space in which the targets are assumed to lie, and $I_S$ is the indicator function for the region (i.e., $I_S(x) = 1$ for $x \in S$ and 0 otherwise). With these assumptions we get
\[
R_{ij}(z^{i1}_i, z^{j2}_j) = \Vol(S) N(z^{i1}_i, z^{j2}_j, V^1_i + V^2_j).
\]
(8)

The standard way of producing the prior volume $\Vol(S)$ is to estimate it from the data. Beyond the theoretical problem this poses in violating Bayesian methodology, there is the practical problem that the presence of outliers can wreak havoc with a volume estimator. A more flexible model is that the tracks are clustered into several regions in which the uniform prior assumption is valid. Therefore one first must cluster tracks, then apply XMAP to each cluster individually. This introduces the requirement that the volume estimator function robustly even on a cluster of very few tracks. This concern is addressed in Section 2.2.

### 2.1 Volume estimator

To construct a representative prior region $S$ we first relabel all measurements from both sensors $\{z_s\}_{s=1}^M$. 


The center of the region $S$ we take to be

$$\hat{x} = \frac{1}{N} \sum_{i=1}^{N} z_i. \quad (9)$$

If the metric data consists of position only, then a sensible shape to prescribe for $S$ is an ellipsoid. To get the shape of this ellipsoid we compute the covariance matrix $\hat{V}$ of the measurements $\{z_i\}$:

$$\hat{V}_{kl} = \frac{1}{N} \sum_{i=1}^{N} (z_{ik} - \hat{x}_k)(z_{il} - \hat{x}_l). \quad (10)$$

One can also compute the covariance matrix of the ellipsoid $S$ by replacing the sum over all $\{z_i\}$ in (10) with an integral over all points in $S$. We choose $S$ to be the (unique) ellipsoid with covariance matrix $\hat{V}$ and centroid $\hat{x}$. The volume of $S$ is

$$\text{Vol}(S) = (2\pi\theta)^{n/2} \sqrt{\text{det}(\hat{V})}, \quad (11)$$

where $n$ is the number of dimensions of the metric data, and

$$\theta = (1 + m/2) (m/2)^{-2/m}. \quad (12)$$

(We reserve the option of choosing $m$ in (12) to be a divisor of $n$ rather than $n$ itself, letting $S$ be the Cartesian product of $m$-dimensional ellipsoids. E.g., we could have $m = 3$ and $n = 6$ if the metric data consists of 3-d position and velocity. In general, we set $m$ to be the number of dimensions of the underlying physical space.) The values of $\theta$ for $m = 1, 2, 3$ are $6/\pi \approx 1.91, 2$, and $(5/3)\sqrt{6/\pi} \approx 2.07$, respectively.

Using this volume estimate we may rewrite (8) as

$$R_{ij}(z_i^1, z_j^2) = \theta^{n/2} \sqrt{\frac{\text{det}(\hat{V})}{\text{det}(\hat{V}_i^1 + \hat{V}_j^2)}} \times \exp\left( -\frac{1}{2} (z_i^1 - z_j^2)^T (\hat{V}_i^1 + \hat{V}_j^2)^{-1} (z_i^1 - z_j^2) \right). \quad (13)$$

There are some problems with (13), however. When $N \leq n$, $\text{det}(\hat{V}) = 0$, so $R_{ij} = 0$, precluding the association of any tracks at all. Furthermore, when $\hat{V}_i^s$ becomes large, we also have $R_{ij} \to 0$, precluding the association of any track with large variance. A solution to the first problem might be to modify the volume estimate somehow so that it always produces positive values, and a solution to the second might be to constrain the covariance matrices $\hat{V}_i^s$ not to extend beyond $S$. But instead of cobbling such ad hoc fixes onto (13), we address the root cause of these problems.

### 2.2 Ellipsoidal method

The standard simplification made in deriving (8) is replacing the integral over $S$ with an integral over all space. This simplification is clearly inappropriate when the variance $\hat{V}_s$ of a track extends beyond the region $S$. Integrating exactly over $S$ is not an appealing alternative. However, we can approximate the integral over $S$ in a manner that respects its finite extent while producing a simple result.

The appeal of approximating $p^0(x)$ by $1/\text{Vol}(S)$ is that this is a homogeneous function: it is invariant under all spatial translations. Only the constant function has this property, so using any other functional form for $p^0(x)$ will disrupt the homogeneous form of the result: e.g., $R_{ij}$ might depend not only on $z_i^1 - z_j^2$, but on the distances of $z_i^1$ and $z_j^2$ from $\hat{x}$ as well. Therefore we approximate $p^0(x)$ not by a single function of $x$, but by a kernel $f(x - c)$, where the shift $c$ translates the kernel to weight the dominant region of the integrand. This is easy to do in (6) and (7) because the integrand is proportional to a Gaussian in each case (the product of two Gaussians being proportional to a Gaussian in the case of (7)), so $c$ is chosen to be the mean of this Gaussian.

For $f$ to have a shape similar to that of $p^0$, the variance of $f$ should be proportional to that of $p^0$, i.e., to $\hat{V}$. In order to get simple results, we choose $f$ to be proportional to a Gaussian. Thus we model $f$ as $f(x - c) = C N(x; c, k\hat{V})$. Two stipulations we place on $f$ is that it yield the correct result in the limits $V_s^1 \gg \hat{V}$ and $V_s^1 \ll \hat{V}$. The requirement implied by the first limit is that $C \leq 1$; the requirement implied by the second is that $f(0) = p^0(0) = 1/\text{Vol}(S)$, which in turn implies $k = \theta$. The resulting formula for $R_{ij}$ is

$$R_{ij}(z_i^1, z_j^2) = \sqrt{\frac{\text{det}(I + M_i^1)}{\text{det}(I + M_i^1 + M_j^2)}} \times \exp\left( -\frac{1}{2} (z_i^1 - z_j^2)^T (V_i^1 + V_j^2)^{-1} (z_i^1 - z_j^2) \right), \quad (14)$$

where $M_i^s = \theta (V_i^s)^{-1} \hat{V}$. Note that (14) does not degenerate for large values of $V_i^s$ or singular matrices $\hat{V}$. Even when a cluster consists of only one or two tracks, (14) gives sensible results, in contrast to (13).

Figure 1: Comparison of approximations

The accuracy of (14) is illustrated by Figure 1, which compares the costs for (a) the traditional approximation of the prior by a constant function and (b) the Gaussian approximation to the cost for (c) the exact result for 1-d data. In this case, $\text{Vol}(S) = 0.6$, and the cost is plotted against the standard deviation of the track error. The Gaussian approximation is accurate across the entire range of tracks errors, whereas the tradition method fails when the track errors are too big relative the given value of $\text{Vol}(S)$.

The effectiveness of this method of computing a threshold depends on how reasonable a fit the uni-
form distribution over some region is to the data. Because such a fit can be fairly poor, a better fit is often produced by clustering the data, and then assigning a uniform prior over each cluster. In six-dimensional position-velocity space, this requires at least seven data points to determine a positive volume. However, the ellipsoidal method described here requires only two points in a cluster. Thus it is completely general (because a one-point cluster is presumably already resolved as not being associated). The performance of such a method would depend on the clustering algorithm employed. It would be useful to compare the performance of this method (for various clustering algorithms) with alternative approaches such as in situ tuning schemes.

3 Type/feature component

In addition to metric information, two other kinds of information often available for data association are target-type and feature. There is a fear associated with incorporating such data into a metric-only association. Metric association is a well established technology, and the improper incorporation of target-type or feature data can lead to serious performance degradation. However, when handled correctly, additional information only improves performance. There are three keys to the incorporating this data properly. First, one must have an appropriate model of the measurement likelihood function for the target-type and/or feature. Second, one must process that likelihood model correctly into a cost. Third, one must account for the possibility of the data not fitting the model. Meeting the first requirement requires access to large experimental or simulated data sets for the feature of interest. The methodology for extracting a likelihood model from such data is not addressed here. What is addressed are the second requirement, which is the topic of the remainder of this section, and the third, which is addressed in Section 3.1.

A target type is a discrete variable representing a finite number of classes to which the target could belong. It is typically extracted (i.e., measured) by a classifier. The possible classification calls c are usually the same as the possible target types t, but for the purposes of data association there is no requirement that the two are related. The quality of a classifier is determined by its confusion matrix. We use $L^s(c|t)$ to denote the confusion matrix entries for sensor $s$.

A feature is typically a single real-valued variable (or perhaps a real array) which is measured by a feature extractor. When the feature is independent of target type, the extractor is characterized by the measurement likelihood function $L^s(w|y)$: the probability density of extracting the value $w$ when the true feature state is $y$. Here, however, we assume that the feature and target type variables are interrelated. We denote the joint target-type/feature state and measurement variables $x^j = x = (t, y)$ and $z^j = z = (c, w)$, respectively.

The joint target-type/feature component in (4) may be written

$$R_{ij}^j = R(z^j_i, z^j_j) = \frac{P^{12}(z^j_1, z^j_2)}{P^1(z^j_1)P^2(z^j_2)},$$

where

$$P^s(z^j_i) = \int P^s_D(x)Q_D^{−1}(x)L^s(z^j_i|x)p^0(x)\,dx,$$

and

$$P^{12}(z^j_1, z^j_2) = \int P^s_D(x)P^s_D(x)\times L^1(z^j_1|x)L^2(z^j_2|x)p^0(x)\,dx.$$  

The measurement likelihood functions do not explicitly depend on the track index here (unlike the metric case) because there is no additional information being stored with the track (such as a covariance matrix $V^s_r$) needed by the measurement likelihood function.

Given the representation $x = (t, y)$ and $z = (c, w)$, we may express the required prior and measurement likelihood functions as

$$p^0(t, y) = p^0(t)p^0(y|t),$$

and

$$L^s(c^i_s, w^i_s|t, y) = L^s(c^i_s|t, y)L^s(w^i_s|c^i_s, t, y).$$

Finally, we assume that the detection probability depends only on the target type:

$$P^D_D(t, y) = P^D_D(t).$$
Using (18), (20), and (21) we may simplify (16) and (17) to
\[ P^w(c^t_i, w^t_i) = \sum_t P^w_D(t) Q^3_D(t) \times \]
\[ L^w(c^t_i | t) P^0(t) P(w^t_i | t), \]
(22)
and
\[ P^{12}(c^t_i, w^t_1, c^t_j, w^t_2) = \sum_t P^w_D(t) P^2_D(t) \times \]
\[ L^1(c^t_i | t)L^2(c^t_j | t) P^0(t) P(w^t_1, w^t_2 | t), \]
where
\[ P(w | t) = \int L(w | t, y) p^0(y | t) dy, \]
(24)
and
\[ P^{w^1, w^2 | t} = \]
\[ \int L(w^1 | t, y) L(w^2 | t, y) p^0(y | t) dy. \]
(25)

It remains to evaluate the integrals in (24) and (25). The first step in doing so involves making a very general assumption about the form of the measurement noise which we call the non-informative noise assumption. This step ensures the robustness of the procedure while producing very simple formulas.

3.1 Non-informative noise

A Gaussian is a typical model for \( L(w | t, y) \), the probability of a target of type \( t \) in feature state \( y \) generating feature measurement \( w \). Such a model carries the risk of returning incredibly tiny assessments of probability for a measurement \( w \) arising (e.g., \( 10^{-100}, 10^{-1000} \) or smaller) when it doesn’t match \( y \) well. In a realistic situation, the probability of a measurement \( w \) could never be that small because there is always the possibility of some glitch in the feature extraction routine. By allowing such tiny probabilities to occur in the model, one runs the risk of the feature extractor completely preventing a pair of tracks being associated even when the metric information is extremely favorable to association. Because one of the chief fears in incorporating a feature extractor into an association algorithm is that it might ruin metric-only performance that is already fairly good, it is quite prudent to account for the possibility of noise in the feature measurement model. (Note: one could make the same argument to point out that the rapid decay of Gaussians allows anomalous metric data to override perfect feature matches, so one might include a noise term in the metric association terms as well. This would be a point worth considering when feature extraction technology reaches the maturity that metric tracking enjoys.)

We regard a function as representing pure noise when the distribution of measurements \( w \) it yields is independent of the actual feature state \( y \). One option for a noise model is a uniform distribution of \( w \) over a certain range. Although this seems simple, it introduces an additional parameter (the width of the distribution) and complicates the required integrals, but is not necessarily a good model of noise. We term the noise model employed in XMAP the non-informative model. This model sets the distribution of measurements due to noise equal to the overall distribution of measurements. It is called non-informative because a measurement \( w \) provides no information as to whether it arose from a clean measurement of some target or merely from noise. Were one to know how noise differs statistically from clean measurements, one could use this information to flag certain measurements \( w \) as more likely to have arisen from noise than others, and perhaps squeeze even more performance out of an association algorithm, but at the risk of algorithm robustness should the noise behave differently than expected. In contrast, the non-informative assumption provides a conservative, robust baseline model for noise.

To use this model, we express the measurement likelihood function as
\[ L(w | t, y) = a^w_{t} L^{\mu}(w | t, y) + (1 - a^w_{t}) P(w | t), \]
(26)
where \( L^{\mu}(w | t, y) \) is the modeled term of \( L(w | t, y) \), and \( a^w_{t} \) is the weight given to this term. The remaining weight \( 1 - a^w_{t} \) is given to the non-informative noise term \( P(w | t) \) defined in (24). (Setting \( a^w_{t} = 0 \) indicates that the feature does not exist or cannot be extracted for target type \( t \).) We define the following analogs of (24) and (25) for \( L^{\mu} \):
\[ P^{\mu}(w | t) = \int L^{\mu}(w | t, y) p^0(y | t) dy, \]
(27)
and
\[ P^{\mu}(w^1, w^2 | t) = \]
\[ \int L^{\mu}(w^1 | t, y) L^{\mu}(w^2 | t, y) p^0(y | t) dy. \]
(28)
Because the distribution of the noise is simply \( P(w | t) \) (which is the distribution the noise and non-noise together), the distribution of the non-noise is also \( P(w | t) \)
\[ P(w | t) = P^{\mu}(w | t), \]
(29)
which can be seen by substituting (26) into (24). The value of \( P(w^1, w^2 | t) \) is a little more interesting:
\[ P(w^1, w^2 | t) = (a^w_{t})^2 P^{\mu}(w^1, w^2 | t) + \]
\[ (1 - (a^w_{t})^2) P^{\mu}(w^1 | t) P^{\mu}(w^2 | t). \]
(30)
A fortuitous consequence of the non-informative noise assumption is that it only slightly complicates the required computations. The quantity \( P(w | t) \) is identical to \( P^{\mu}(w | t) \), whereas \( P(w^1, w^2 | t) \) is a simple combination of \( P^{\mu}(w^1, w^2 | t), P^{\mu}(w^1 | t) \), and \( P^{\mu}(w^2 | t) \). To complete the formulation, we now compute these \( P^{\mu} \) quantities for a representative measurement likelihood model.

3.2 Result

The final step of the derivation is modeling the feature prior as
\[ p^0(y | t) = \frac{1}{\sum I_{\Delta_{t}}(y)}, \]
(31)
where \( I_\Delta \) is the indicator function over an interval of length \( \Delta \). We employ the standard simplification of extending integrals over this interval to the entire real line, which is valid provided \( \Delta \) is fairly large compared to the measurement error defined by \( L^*(w|t, y) \). This simplifies (27) and (28) to

\[
P^\mu(w|t) = (\Delta_t)^{-1},
\]

(32)

\[
P^\mu(w^1, w^2|t) = (\Delta_t)^{-1} K_t(w^1, w^2),
\]

(33)

where

\[
K_t(w^1, w^2) = \int L^\mu(w^1|t, y) L^\mu(w^2|t, y) \, dy.
\]

(34)

We now collect the equations above into a single, simple result. The goal is a formula for \( R^j_t \), which is given by (15). This may be simplified to

\[
R^j_t = R(c^1_t, w^1_t, c^2_t, w^2_t) = \frac{P^{12}(c^1_t, w^1_t, c^2_t, w^2_t)}{P^1(c^1_t) P^2(c^2_t)},
\]

(35)

where (22) and (23) have been simplified to

\[
P^\ast(c) = \sum_t P^0(t) P_D(t) Q_{\frac{1}{2}^+}^w(t) L^x(c|t),
\]

(36)

and

\[
P^{12}(c^1, w^1, c^2, w^2) = \sum_t \frac{P^0(t)}{\Delta_t} P_D(t) P_N^w(t) L^x(c^1|t) L^w(c^2|t) \times
\]

(37)

\[
\left(1 - \frac{(a^\mu)^2}{\Delta_t} + (a^\mu)^2 K_t(w^1, w^2)\right),
\]

respectively.

Despite the simplicity of the equations (35)–(37), they are quite powerful, for they indicate the proper way to combine the prior distribution of target type \( p^0(t) \), the detection probabilities \( P_D^2(t) \) for each sensor and target type, the prior spread of the feature value \( \Delta_t \) for each type, and the weight \( a^\mu_t \) to assign to the model for each type, with the confusion matrices \( L^x(c|t) \) for each sensor, and the kernel \( K_t(w^1, w^2) \) that incorporates the feature measurement model for each target type \( t \). This may be more freedom than one desires, in which case the prior on target type \( p^0(t) \) could be set to a uniform distribution, the detection probabilities \( P_D(t) \) to a single value \( P_D \), and the prior feature spread to a single value \( \Delta \). The values \( L^x(c|t) \), on the other hand, are a property of the classifier, and are determined when the classifier is calibrated. Similarly the values of \( a^\mu_t \) and \( K_t(w^1, w^2) \) are properties of the feature extractor. We now give an example of \( K_t(w^1, w^2) \) for a specific, hypothetical situation.

### 3.3 Example: bimodal feature model

The feature model should be determined from Monte Carlo simulations or similar studies of the feature extractor over a range of realistic conditions. To give a concrete example, we consider a hypothetical example in which it is observed that for every target type \( t \), the measurements often cluster near the true feature value, but sometimes cluster about a constant offset from the true value. In this hypothetical situation, we find: that the offset \( \delta \) is the same for every target type; that the variance about the true and offset values are roughly the same as each other, but vary with \( t \); and that the probabilities of falling near the true value, offset value, or elsewhere vary with \( t \) as well. In such a situation, the following measurement likelihood model would be appropriate:

\[
L^\mu(w|t, y) = C_0^0 \mathcal{N}(w; y, V_t) + C_1^1 \mathcal{N}(w; y + \delta, V_t),
\]

(38)

where \( C_0^0 + C_1^1 = 1 \). The empirical study would determine all of the constants involved, as well as \( a^\mu_t \). E.g., \( a^\mu_t \) represents the fraction of measurements falling near the true or the offset value, etc. (Assuming \( V_t < \Delta^2 \), it is valid to use apply the simplification of extending integrals over the entire real line, which was invoked in derivation of (35).)

Using this bimodal model, we find

\[
K_t(w^1, w^2) = \left(A^0_t(\mathcal{N}(\Delta w; 0, 2V_t) + A^1_t \left(\mathcal{N}(\Delta w; \delta, 2V_t) + \mathcal{N}(\Delta w; -\delta, 2V_t)\right)\right),
\]

(39)

with \( \Delta w = w^1 - w^2 \), \( A^0_t = (C_0^0)^2 + (C_1^1)^2 \), and \( A^1_t = C_0^0 C_1^1 \). This completes the explicit specification of the probability ratio \( R^j_t \) for the joint target-type/feature component of the XMAP method.

### 4 Conclusions

The XMAP procedure for track-to-track association is able to utilize any kind of data in a mathematically rigorous way provided one is able to model the prior and measurement likelihood functions for the data. Section 2 develops the metric component of XMAP, which is similar to the standard formulation, but uses a different adaptive threshold which properly reflects the finite variance of the prior distribution of the metric state. Section 3 shows how to incorporate joint feature/target-type data in a manner that exploits the confusion matrices typically provided for a target-type classifier, while providing the user with the opportunity to obtain improved association results by formulating measurement models for the feature extractor for each target type \( t \). Typically, one would use a single model for groups of target types with similar behavior—the power of this formulation comes from being able to use the classifier to distinguish between gross differences in behavior of the feature measurement process. Because the probabilistic structure of the data is incorporated correctly, there is no danger of performance degradation versus metric-only association, provided one provides the correct measurement likelihood functions and confusion matrices for the data. In particular, the non-informative noise assumption lends great robustness to the results and produces simple formulas as well.
A Probability ratio derivation

A.1 Single sensor

In this section we compute the probability $P(z^*, a^s | x, n)$ of getting a specific array of measurement values $z^*$ on a sensor $s$, and getting them in the way prescribed by the target-to-track map $a^s$, given that there are $n$ targets whose states are listed in the array $x$.

To begin, we assume there is a finite set $J = \{1, 2, \ldots, n\}$ of targets whose states are given by $x = \{x_1, x_2, \ldots, x_n\}$. The probability of a sensor $s$ detecting a target in state $x$ is denoted $P^s_{D}(x)$, and $Q^s_{D}(x) = 1 - P^s_{D}(x)$. For the given sensor $s$, the probability of detecting a target $j$ is assumed to depend only on the state $x_j$ of the target. Given these assumptions, the probability of the subset of targets detected by sensor $s$ being $J^s$ is

$$P(J^s | x, n) = \prod_{j \in J^s} P^s_{D}(x_j) \prod_{j \in J \setminus J^s} Q^s_{D}(x_j). \quad (40)$$

We assume that each detected target produces exactly one track on the sensor (no split or merged tracks), and that tracks cannot arise without targets producing them (no false alarms). Thus there is a one-to-one correspondence between the set of detected targets $J^s$ and the tracks on sensor $s$. Letting $n^s = |J^s|$, we define $I^s = \{1, 2, \ldots, n^s\}$ to be the track indices, and let $a^s : J^s \rightarrow I^s$ denote a bijection from targets to tracks. We assume that all such bijections are equally likely, i.e., that $P(a^s | J^s, x, n) = 1/n^s!$. Multiplying this expression by $(40)$ yields an expression for $P(a^s | x, n)$, which may be written more simply as $P(a^s | x, n)$, since $a^s$ determines $J^s$:

$$P(a^s | x, n) = \frac{1}{n^s!} \prod_{j \in J^s} P^s_{D}(x_j) \prod_{j \in J \setminus J^s} Q^s_{D}(x_j). \quad (41)$$

Now let $z^*_i$ denote the measurement on track $i$ on sensor $s$, and $z^*$ the array of all measurements. Similarly, let $L^s_i(z^*_i|x)$ denote the measurement likelihood function for track $i$ on sensor $s$: i.e., the probability (or probability density) of a detected target in state $x$ producing the measurement $z^*_i$. Given the $n$ states $x$, and the map which assigns a subset of them to the tracks on sensor $s$, the probability of getting the array of measurements $z$ is

$$P(z^* | a^s, x, n) = \prod_{j \in J^s} L^s_{a^s(j)}(z^*_a(j) | x_j). \quad (42)$$

Multiplying (41) and (42) yields the desired quantity:

$$P(z^*, a^s | x, n) = \frac{1}{n^s!} \times \prod_{j \in J^s} P^s_{D}(x_j) L^s_{a^s(j)}(z^*_a(j) | x_j) \prod_{j \in J \setminus J^s} Q^s_{D}(x_j). \quad (43)$$

A.2 Multiple sensors

Is is straightforward to generalize (43) from a single sensor to a set of sensors $S$. We let $z = (z^*)_{s \in S}$ denote the array of all individual measurement arrays $z^*$, and similarly let $a = (a^s)_{s \in S}$ denote all target-to-track maps. Assuming the measurement process is independent for each sensor, we have

$$P(z, a | x, n) = \prod_{s \in S} \left( \frac{1}{n^s!} \times \prod_{j \in J^s} P^s_{D}(x_j) L^s_{a^s(j)}(z^*_a(j) | x_j) \prod_{j \in J \setminus J^s} Q^s_{D}(x_j) \right). \quad (44)$$

Although $a$ defines which targets produce which tracks, it is more convenient to express the information contained in $a$ differently. We define

$$a(j) = \{(s, i) : a^s(j) = i\}, \quad (45)$$

$$a_S(j) = \{s : j \in J^s\}. \quad (46)$$

The function $a_S$ gives the set of sensors which detect each target $j$, whereas $a$ gives the track indices of each detection too. In this new notation we express (44) as

$$P(z, a | x, n) = \left( \prod_{s \in S} \frac{1}{n^s!} \right) \prod_{j \in J} \left( \prod_{(s, i) \in a(j)} P^s_{D}(x_j) L^s_{i}(z^*_i | x_j) \prod_{s \in S \setminus a(j)} Q^s_{D}(x_j) \right). \quad (47)$$

We may eliminate the dependence on $x$ from (47) by integrating over the product of (47) and $p^0(x|n)$. We assume that the individual target states are i.d. so $p^0(x|n)$ is the product of $p^0(x|n)$ over $j \in J$. Hence

$$P(z, a | n) = \int \left( \prod_{j \in J} p^0(x_j) \right) P(z, a | x, n) \, dx. \quad (48)$$

This integral over $x$ separates into a product of integrals over each $x_j$. These integrals are particularly
simple for targets \( j \) that are not detected by any sensor; each such integral is equal to

\[
q = \int p^0(x) \prod_{s \in S} Q^s_D(x) \, dx .
\]  

(49)

We let

\[
J_D = \{ j \in J : |a(j)| \geq 1 \}
\]

(50)

be the set of detected targets, and \( n_D = |J_D| \) be the number detected. We may now simplify (48) to

\[
P(\mathbf{a}|n) = \left( \prod_{s \in S} \frac{1}{n^s!} \right) q^{n-n_D} \prod_{j \in J_D} \int p^0(x) \times
\]

\[
\prod_{(s,i) \in a(j)} P^s_D(x) L^s_i(z^s_i|x) \prod_{s \in S \setminus a(j)} Q^s_D(x) \, dx .
\]  

(51)

A.3 Association probability

The function \( a \) defined in (45) maps each target to the cluster of tracks it produces. An association \( [a] \) is defined to be a set of such clusters,

\[
[a] = \{ a(j) : j \in J_D \} .
\]

(52)

Different functions \( a \neq a' \) can yield the same association: i.e., \( a = [a'] \). Indeed, there are exactly \( n!/n^i (n-n_D)! \) functions which yield the association \( [a] \), all of which are equally probable, so

\[
P(\mathbf{a}|n) = \frac{1}{(n-n_D)!} P(\mathbf{a}|n) .
\]

(53)

If the prior probability for the number of targets being \( n \) is denoted \( \rho^0(n) \), then

\[
P(\mathbf{a}|n) = \sum_{n=0}^{\infty} \rho^0(n) P(\mathbf{a}|n) .
\]

(54)

From (54), the association probability follows immediately, because \( P([a]|\mathbf{z}) = P(\mathbf{a}|[a])/P(\mathbf{z}) \). We define

\[
\gamma(n_D,q) = \sum_{n=0}^{\infty} \rho^0(n) \frac{n!}{(n-n_D)!} q^{n-n_D} ,
\]

(55)

and write

\[
P([a]|\mathbf{z}) = \left( \frac{\gamma(n_D,q)}{P^{\mathbf{z}}} \prod_{s \in S} \frac{1}{n^s!} \right) \prod_{j \in J_D} \int p^0(x) \times
\]

\[
\prod_{(s,i) \in a(j)} P^s_D(x) L^s_i(z^s_i|x) \prod_{s \in S \setminus a(j)} Q^s_D(x) \, dx .
\]  

(56)

A.4 Poisson prior case

We will now re-write (56) as a constant leading factor (i.e., a quantity independent of \([a]\)) times a product over only those targets that produce at least two tracks,

\[
J^+_D = \{ j \in J : |a(j)| \geq 2 \} .
\]

(57)

Such a form allows the use of efficient algorithms such as JVC [6] to find the association with maximal probability in the two-sensor case.

The first step is to re-write (56) to have a constant leading factor. (The leading factor in (56) depends on \( n_D \), which in turn depends on \([a]\).) One can accomplish this by assuming a Poisson prior \( \rho^0(n) = e^{-\nu} \nu^n / n! \) on the number of targets, where \( \nu \) is the prior expected number of targets. (Indeed, one cannot get a constant leading factor without this assumption [4].) In this case, (55) simplifies to

\[
\gamma(n_D,q) = e^{(q-1)\nu} \nu^{n_D} ,
\]

(56)

which allows (56) to be written as

\[
P([a]|\mathbf{z}) = \left( \frac{e^{(q-1)\nu}}{P^{\mathbf{z}}} \prod_{s \in S} \frac{1}{n^s!} \right) \prod_{j \in J_D} \int p^0(x) \times
\]

\[
\prod_{(s,i) \in a(j)} P^s_D(x) L^s_i(z^s_i|x) \prod_{s \in S \setminus a(j)} Q^s_D(x) \, dx .
\]  

(58)

The second step is to divide each side of (58) by \( P([a_0]|\mathbf{z}) \), where \([a_0]\) is the null association, which maps every target in a set \( J^+_D \) to a to a cluster containing only a single track, i.e., \([a_0(j)] = 1 \) for all \( j \in J^+_D \). The constant leading factors cancel, and we are left with the ratio of a product over \( J_D \) and a product over \( J^+_D \). We simplify this ratio as follows. For each cluster of tracks \( \alpha \in [a] \) we collect the factors of \( P([a_0]|\mathbf{z}) \) corresponding to the tracks in \( \alpha \) (i.e., those with \( j \in J^+_D \) such that \( a_0(j) \subseteq \alpha \)) to define

\[
P([a]|\mathbf{z}) = P([a_0]|\mathbf{z}) \prod_{j \in J^+_D} R^\alpha(j)(\mathbf{z}) .
\]

(59)

In the two-sensor case, any cluster of at least two tracks may be expressed in the form \( \alpha = \{(i, 1), (j, 2)\} \), and (59) may be written

\[
R^\alpha(j) = \frac{\nu P^1_D(z^1_j, z^2_j) P^2_D(z^2_j)}{\nu P^1_D(z^1_j) P^2_D(z^2_j)} ,
\]

(61)

where

\[
P^s_D(z^s_i) = \int p^0(x) P^s_D(x) Q^{s-2}(x) L^s_i(z^s_i|x) \, dx ,
\]

(62)

and

\[
P^{12}_{ij}(z^1_i, z^2_j) = \int p^0(x) P^1_D(x) P^2_D(x) L^1_i(z^1_i) L^2_j(z^2_j) \, dx .
\]

(63)

Equations (61)–(63) are the starting point of the derivation in the main body of the paper, though they are written is a slightly different form there to facilitate further manipulations.