A tracking algorithm based on classification information

Li Ren and Zuzong Muo
University of electronic science and technology of china, Chengdu, China

Abstract. With the development of the sensor, the sensor can get more information of targets. Therefore, Scholars start to research new information of targets and a variety of methods are emerged in the process of exploring. It is meaningful for multiple targets tracking that if the sensors can distinguish target classes. In this paper, we study how to use the target classification information to support multiple targets tracking, and we make a further improvement on the JPDA algorithm. Comparison is made in simulation to compare different methods.

Keywords: multiple targets tracking; classification information; data association.

1 Introduction

After years of development, multiple targets tracking has made many achievements. In data association NN (Nearest neighbor) is the easiest algorithm. But NN can not solve complicate problems. So PDA (probabilistic data association) is proposed to solve this problems. As PDA can only track single target, JPDA (joint probabilistic data association) is proposed to track multiple targets. Based on this, reference [5] proposed a tracking method that is based on classification information. And reference [6] apply this method in JPDA. IMMJPDA (interactive multiple-model joint probabilistic data association) is an excellent method for multiple targets tracking. And CIMMJPDA [7] (comprehensive IMMJPDA) is proposed to solve the complication of IMMJPDA. In this thesis, we study a classification-aided algorithm in multiple targets tracking base on CIMMJPDA.

2 Confusion matrix definition

Classification-aided is first proposed in reference [5], Bar-shalom use confusion matrix to denotes the accuracy of sensor.

Assume that target classes is N, \( \zeta \) is the output of classifier. So \( \zeta \) can be written as

\[
\zeta \in \{1, 2, ..., N\}
\]

The confusion matrix is

\[
C = [c_{nm}]
\]
Where \( c_{nm} = p(\zeta = m | k = n)\), \( n, m = 1, 2, ..., N \) is the likelihood of the true class being \( n \) when the classifier output is \( m \). Each column represent the probabilities of classifier’s output is \( m \). So when the output is \( m \), the post probability of the target being class \( n \) is

\[
\mu_n = p(k = n | \zeta = m) = \frac{c_{nm}\mu_n^0}{\sum_{i=1}^{N} c_{im}\mu_i^0}
\]  

(3)

Where \( \mu_n^0 \) is the prior probability of the target being class \( n \).

Similarly, when considering the tracking process, along with the time \( k \) constantly moving forward, so it is assumed that the true target probability was \( \bar{\mu}(k-1) \) (cumulative output). According to the recurrence of time and the above formula, we can know that \( \bar{\mu}^0 \) actually is similar to the \( \bar{\mu}(k-1) \), so we get the recursive representation of the time sequence

\[
\bar{\mu}(k) = \frac{\bar{c}_j \otimes \bar{\mu}(k-1)}{\bar{c}_j \bar{\mu}(k-1)}
\]  

(4)

Where \( \otimes \) is the Schur-Hadamard product, \( \bar{\mu}(k-1) \) denotes the cumulative classification information at time \( k-1 \). So \( \bar{\mu}(0) \) is the initial value.

3 JPDA data association

3.1 Confirm matrix definition

Bar-Shalom define a validation matrix as follows

\[
\Omega = \left[ \bar{\omega}_j \right] \quad j = 1, ..., m_k ; t = 0, 1, ..., T
\]  

(5)

Where \( T \) is the tracks number, \( m_k \) is total measurements, Index \( t = 0 \) represents that the measurement is a false alarm or a complex wave, \( \bar{\omega}_j \) is binary element, and when it is 1 stands for measurement \( j \) lies in the validation gate for target \( t \).

3.2 Split validation matrix

After generating validation matrix, we will split validation matrix to generate feasible matrices. Assume that joint event number is \( n_k \). In joint event \( i \) ( \( i = 1, 2, ..., n_k \) ), \( \theta_i^j(k) \) stands for measurement \( j \) lies in the validation gate for target \( t (0 \leq t \leq T) \).

We define \( \hat{\Omega}[\theta_i^j(k)] \) as joint event \( i \), so we have

\[
\hat{\Omega}[\theta_i^j(k)] = \left[ \hat{\omega}_j^i(\theta_i^j(k)) \right] \quad j = 1, ..., m_k ; t = 0, 1, ..., T; i = 1, ..., n_k
\]  

(9)

Where \( \hat{\omega}_j^i(\theta_i^j(k)) = 1 \) stands for measurement \( j \) lies in the validation gate for target \( t \).

Split a validation matrix must follow two principles:

1) There is only one nonzero element in every row of feasible matrix after splitting validation matrix;
(2) There is not more than one nonzero element in every column of feasible except the first column.

### 3.3 Joint event probability calculation

After the joint event, the probability is needed to be calculated. Here we define three binary elements: \( \phi[\theta(k)] \) stands for false target numbers in joint event \( \theta(k) \), when \( \tau_j[\theta(k)] = 1 \) stands for a measurement lie in at least one target’s validation gate, when \( \delta_j[\theta(k)] = 1 \) stands for there at least one measurement lie in the validation gate for target \( t \). When there is a measure \( j \) is associated to actual object in joint event \( \theta(k) \), \( \tau_j[\theta(k)] = 1 \), relatively, if \( \delta_j[\theta(k)] = 1 \), The object \( t \) has the correlation with the measurement.

\[
\tau_j[\theta(k)] = \sum_{j=1}^{m} \delta^j_j[\theta(k)] = \begin{cases} 
1 \\
0 
\end{cases} 
(6)
\]

\[
\delta_j[\theta(k)] = \sum_{j=1}^{m} \delta^j_j[\theta(k)] = \begin{cases} 
1 \\
0 
\end{cases} 
(7)
\]

\[ \phi[\theta(k)] = \sum_{j=0}^{m} [1 - \tau_j[\theta(k)]] \]

(8)

Then using Bayesian rule, Conditional probability of Joint event \( \theta(k) \) can be written as follows

\[
\Pr\{\theta(k) \mid Z^k\} = \Pr\{\theta(k) \mid Z(k), Z^{k-1}\} = \frac{1}{c} \Pr\{Z(k) \mid \theta(k), Z^{k-1}\} \Pr\{\theta(k)\}
\]

Where \( c \) is normalization constant, if the measured value of the target is subject to the Gauss distribution, that is

\[
N_j[Z_j(k)] = N[Z_j(k); \hat{Z}_j(k, |k-1), S_j(k)]
\]

\( S_j(k) \) stands for the new covariance matrix of the target \( t \). \( \hat{Z}_j(k, |k-1) \) is the forecast value measured by the target \( t \) at time \( k \). Generally in the volume \( V \) of the region, relatively measurements from false target are evenly distributed.

In the joint event \( i \), the joint probability of the measurement \( j \) and real track \( t \)

\[
p[Z_j(k) \mid \phi^j_j, Z^{k-1}] = N_j[Z_j(k)]
\]

\[
= \frac{1}{\sqrt{2\pi S_j(k)}} \exp\left[-\frac{1}{2} (Z_j(k) - \hat{Z}_j(k))'S_j^{-1}(k)(Z_j(k) - \hat{Z}_j(k))\right]
\]

(11)

In the joint event \( i \), the probability of the measurement \( j \) and clutter, etc

\[
p[Z_j(k) \mid \theta^j_j(k), Z^{k-1}] = V^{-1}
\]

(12)

Finally, Joint event \( \theta(k) \)'s probability can be computed as follows at time \( k \)

\[
\Pr\{\theta(k) \mid Z^k\} = \frac{1}{c'} \phi[\theta(k)]! \prod_{j=1}^{m} N_j[Z_j(k)]^{\tau_j[\theta(k)]} \prod_{j=2}^{T} (P_{d}^{j[\theta(k)]})^{\delta_j[\theta(k)]}(1 - P_{d}^{j[\theta(k)]})^{1 - \delta_j[\theta(k)]}
\]

(13)
Where $c'$ is a normalization constant, $P_D^t$ stands for the detection probability of the target $t$.

### 3.4 Confirmation of joint probability

From the above three parts, we have been able to calculate the correlation probability of each measurement and each target, and define $\beta_j(k)$ as the joint probabilities of measurement $j$ and target $t$ at time $k$:

$$\beta_j(k) = \Pr\{\theta_j(k) | Z^k\} = \sum_{i=1}^{N_t} \delta_j^i(\theta_i(k)) \Pr\{\theta_i(k) | Z^k\}$$

(14)

### 4 Modified algorithms for classified information

In reference[6] a modified JPDA with classification-aided is proposed to modified $\beta_j(k)$. When several targets are very close each other, the computed $\beta_j(k)$’s error is large, so there is not much significance in revise. In this thesis, we proposed a new modified JPDA which correct the Joint event’s probability.

In joint event $\theta^j(k)$, when a measurement’s output of classifier is $j$. Here we define measurement $m$. So the modified probability density function of the measurement $j$ and target $t$ can be as follows:

$$p_c\{z_m(k) | \theta_m^j(k), Z^{k-1}\} = P_c(k,m,t)$$

$$= P(k,m,t) P_c(k,m,t)$$

$$= p[z_m(k) | \theta_m^j(k), Z^{k-1}] P_c(k,m,t)$$

$$= N_c[Z_n(k)] P_c(k,m,t)$$

$$= \frac{1}{\sqrt{2\pi S_n(k)}} \exp\left\{ -\frac{1}{2} [Z_n(k) - \bar{Z}_n(k)] S_n^{-1}(k) [Z_n(k) - \bar{Z}_n(k)] \right\} \frac{c_j \mu'(k-1)}{c_j \mu^e}$$

(15)

If measurement $m$ is related to a false alarm:

$$p_c\{z_m(k) | \theta_m^e(k), Z^{k-1}\} = P(k,m,t) = V^{-1} \frac{c_j \mu^0}{c_j \mu^e}$$

(16)

Then the probability of joint event $\theta^j(k)$ of modified JPDA is

$$p_c\{\theta^j(k) | Z^k\} = \frac{1}{c} \Pr\{\theta^j(k) | Z^k\} \prod_{n=1}^{M} \left( \frac{c_j \mu^0}{c_j \mu^e} \right)^{1-\omega_n(\theta^j(k))} \prod_{n=1}^{M} \left( \frac{c_j \mu'(k-1)}{c_j \mu^e} \right)^{\omega_n(\theta^j(k))}$$

(17)

$$= \frac{1}{c} \Pr\{\theta^j(k) | Z^k\} \prod_{n=1}^{M} \left( \frac{c_j \mu^0}{c_j \mu^e} \right)^{1-\omega_n(\theta^j(k))} \left( \frac{c_j \mu'(k-1)}{c_j \mu^e} \right)^{\omega_n(\theta^j(k))}$$

Since $\prod_{n=1}^{M} \left( \frac{c_j \mu^0}{c_j \mu^e} \right)$ is the same in each joint event, we can get a new normalization constant.

96
\[ P\left(\Theta_i(k) \mid Z^k\right) = \frac{1}{c^i} \Pr\left(\Theta_i(k) \mid Z^k\right) \prod_{n=1}^{M} \left( \frac{C_n^i \mu_n^i(k-1)}{C_n^i \mu_n^0} \right)^{r_n(\Theta_i(k))} \] \hspace{1cm} (18)

\( c^i \) is a new normalization constant.

### 5 Simulation analyses

In this thesis, in order to verify the tracking performance of the algorithm, there setting a simulation scene, simulation scene is an area of 3000*4000m\(^2\). Detection probability is 0.9 and SNR is 10dB. Pg is 0.9997. Clutter parameter is \( \lambda = 1 \times 10^{-6} \). The kinematic parameters of the three targets are the same. The velocity of target is 15M/s and the turning angular velocity is consistent. The target type for the scene setting is 4 classes, that is to say, the three objectives are divided into one of the four categories. In order to track the maneuvering target, the CIMMJPDA algorithm is chosen as the basic framework in this thesis.

The confusion matrix is defined as

\[ C = \begin{bmatrix} 0.85 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.85 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.85 \end{bmatrix} \hspace{1cm} (19) \]

From the confusion matrix, we can see that the correct classification rate of classifier in this simulation is 0.85.

The prior probability of target class is set to

\[ \mu^0 = \begin{bmatrix} 0.82 & 0.06 & 0.06 \\ 0.06 & 0.82 & 0.06 \\ 0.06 & 0.06 & 0.82 \\ 0.06 & 0.06 & 0.06 \end{bmatrix} \hspace{1cm} (20) \]

On the generation of the clutter class, the classification results of the clutter are set to one of the four categories of uniform distribution. The simulation times of Monte Carlo is 100.

The simulation scene is show in Figure 1.

![Simulation scene](image)

**Figure 1.** Multiple targets simulation scene.

In order to compare the results of the algorithm and the modified algorithm, we get a statistic association results of CIMMJPDA-C and CIMMJPDA-MC, which represent the rate of a real target’s measurement associated to the adjacent tracks.
Figure 2. Mean measurement distribution of target1 (CIMMJPDA-C).

Figure 3. Mean measurement distribution of target1 (CIMMJPDA-MC).

Figure 4. Mean measurement distribution of target2 (CIMMJPDA-C).
Figure 5. Mean measurement distribution of target2 (CIMMJPDA-MC).

Figure 6. Mean measurement distribution of target3 (CIMMJPDA-C).

Figure 7. Mean measurement distribution of target3 (CIMMJPDA-MC).
From the figures, we can see that CIMMJPD-MC has a certain increase in the correct correlation, which shows that the modified algorithm has the effect of improving the correlation probability.

Table 1. Three targets simulation results

<table>
<thead>
<tr>
<th>target</th>
<th>algorithm</th>
<th>RMSE (X)</th>
<th>RMSE (Y)</th>
<th>Lost rate(%)</th>
<th>time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIMMJPD</td>
<td>28.40</td>
<td>39.80</td>
<td>87.5</td>
<td>0.0095</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-C</td>
<td>18.02</td>
<td>20.46</td>
<td>9</td>
<td>0.0099</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-MC</td>
<td>17.18</td>
<td>19.06</td>
<td>4</td>
<td>0.0104</td>
</tr>
<tr>
<td>2</td>
<td>CIMMJPD</td>
<td>32.92</td>
<td>39.74</td>
<td>75.5</td>
<td>0.0095</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-C</td>
<td>18.30</td>
<td>19.50</td>
<td>9</td>
<td>0.0099</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-MC</td>
<td>16.59</td>
<td>16.64</td>
<td>5</td>
<td>0.0104</td>
</tr>
<tr>
<td>3</td>
<td>CIMMJPD</td>
<td>24.37</td>
<td>27.85</td>
<td>81</td>
<td>0.0095</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-C</td>
<td>16.62</td>
<td>18.02</td>
<td>7</td>
<td>0.0099</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-MC</td>
<td>16.18</td>
<td>17.58</td>
<td>3</td>
<td>0.0104</td>
</tr>
</tbody>
</table>

As can be seen from above, when the distance between targets is relatively small, CIMMJPD can hardly distinguish targets, and the rate of loss is very high. The modified algorithm CIMMJPD-C is better than that. The proposed algorithm in this thesis is better than CIMMJPD-C.

For further comparison, we set another simulation scene. In this scene the tracks of three targets are crossed as shown in Figure 8.

Figure 8. Three crossed run targets.

Similarly, when setting the number of simulations to 100. Finally, the simulation results of the three algorithms are shown in Table 3.

Table 2. Three crossed run targets simulation results

<table>
<thead>
<tr>
<th>Target</th>
<th>algorithm</th>
<th>RMSE(X)</th>
<th>RMSE (Y)</th>
<th>Lost rate(%)</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIMMJPD</td>
<td>34.96</td>
<td>36.073</td>
<td>80</td>
<td>0.0097</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-C</td>
<td>19.42</td>
<td>18.98</td>
<td>22</td>
<td>0.0105</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-MC</td>
<td>18.96</td>
<td>18.76</td>
<td>13.5</td>
<td>0.0111</td>
</tr>
<tr>
<td>2</td>
<td>CIMMJPD</td>
<td>32.18</td>
<td>34.23</td>
<td>66</td>
<td>0.0097</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-C</td>
<td>20.69</td>
<td>20.3</td>
<td>14</td>
<td>0.0105</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-MC</td>
<td>18.71</td>
<td>17.87</td>
<td>5</td>
<td>0.0111</td>
</tr>
<tr>
<td>3</td>
<td>CIMMJPD</td>
<td>22.94</td>
<td>26.64</td>
<td>68.5</td>
<td>0.0097</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-C</td>
<td>14.61</td>
<td>17.12</td>
<td>2</td>
<td>0.0105</td>
</tr>
<tr>
<td></td>
<td>CIMMJPD-MC</td>
<td>14.19</td>
<td>16.15</td>
<td>1.5</td>
<td>0.0111</td>
</tr>
</tbody>
</table>
As can be seen from the above results, when targets are cross each other, the proposed algorithm in this thesis can get a better performance.

6 Conclusions

Traditional multiple targets tracking technology is only concerned with the estimation of kinematic parameters, while ignoring the application of other information. Based on the classification ability of the sensor, this thesis apply the classification information into the process of multiple target tracking. The simulation results show that the modified algorithm has better tracking performance under the environment of multiple dense targets, which is important for multiple targets tracking.

References