国 防 科 技 大 学 学 报
第20卷 第3期 JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY Vol.20 No.3 1998

快速 JPDA 算法的递归和并行实现*

程洪玮 周一宇 孙仲康

(国防科技大学电子工程学院 长沙 410073)

摘 要 本文从联合关联事件的构造出发，讨论关联假设事件的分层构造，以达到降低计算量的目的。这里的层次可从0取到 L，0 层表示没有任何目标能够跟当前的观测数据关联，L 层表示共有 L 个目标可以跟当前扫描得到的观测数据相关联。本文的关联事件的构造中，各层次的搜索具有递归性并可以独立进行，因而可以并行实现。文中还将本文的方法跟有关文献作了比较，并且给出相应的计算机仿真实验及其结果。

关键词 概率，数据关联，目标跟踪，快速算法

分类号 O211

Recursive and Parallel Implementation of Fast JPDA Algorithm

(Institude of Electronic Engineering, NUDT, Changsha, 410073)

Abstract This paper discusses a method to reduce calculation load by means of layered searching construction of association hypothesis events. The layer here can take 0 value or an integer L. Layer 0 means that no target can find its association data from current returns. Layer L means that L targets can find their association data from current returns. Our layered construction method in the paper is recursive and of independence among layers, so it can also be implemented in parallel structure. Comparative analysis of the method with relative methods in other references and the corresponding computer simulation tests and the results are also given in the paper.

Key words probability, data association, target tracking, fast algorithm

联合概率数据关联算法(Joint Probabilistic Data Association, JPDA)是密集杂波环境下的一种良好的多目标关联与跟踪的优化算法。但是，当目标的数目增大时，由于目标与观测数据之间的关联假设事件的数量呈指数增长，JPDA 算法的计算量也将呈指数增长甚至造成计算量爆炸，同时还增加了构造联合关联事件的复杂性。为降低计算量，不少文献讨论了次优 JPDA 快速算法，但都以降低关联跟踪性能为代价。Fitzgerald[1] 避开了关联事件的产生问题，提出一个实用经验公式来计算后验概率 β_j^i. 严格地说，Fitzgerald 的方法是有偏的。Roecker[2] 提出了一种次优 JPDA 方法，这种方法建立在目标探测概率为1的假设基础上，也避开了关联假设事件产生的问题。Zhou[3] 在 James[4] 方法的基础上提出了一种深度优先搜索法(Depth-First Search, DFS)来产生关联事件，但其后验概率 β_j^i 的计算无规律可言，该文只好将后验概率的计算分成直接法和近似法进行。

本文将从联合关联事件的构造出发，讨论关联假设事件的分层构造方法，降低 JPDA 方法的计算量。这里的层次可从0取到 L，0 层表示没有任何目标能够跟当前的观测数据关联，L 层表示共有 L 个目标可以跟当前得到的观测数据相关联。

1 关联事件的分层构造及快速算法

设 m^k 为当前扫描(第 k 次扫描)获得的回波个数，T^k 为目标个数。在 JPDA 算法中，第 j 个回波与

* 国防预研基金资助项目
1997年12月5日收稿
第一作者：程洪玮，男，1964年生，博士生

第 t 个目标之间的关联概率 β_j^t 的计算通常是从关联聚矩阵 Ω的构造开始的。由 m_k 个回波和 T_k 个目标构成的关联聚矩阵 Ω是一个 $m_k \times (T_k+1)$ 的行列式[5~7]:

$$\Omega = [\omega_{jt}] = \begin{matrix} \\ 0 \\ 1 \\ 2 \\ \\ m_k \end{matrix} \overbrace{\begin{matrix} 0 & 1 & 2 & \cdots & T_k \\ 1 & \omega_{11} & \omega_{12} & \cdots & \omega_{1T_k} \\ 1 & \omega_{21} & \omega_{22} & \cdots & \omega_{2T_k} \\ & & & & \\ 1 & \omega_{m_k1} & \omega_{m_k2} & \cdots & \omega_{m_kT_k} \end{matrix}}^{t} \begin{matrix} \\ 1 \\ 2 \\ \\ m_k \end{matrix} \Big\}_j \tag{1}$$

式中第一列所有元素 $\omega_{j0}=1$, 指所有的回波都可能源自杂波。其他元素 $\omega_{jt}=1$ $(t\neq 0)$ 指第 j 回波落入了目标 t 的选通波门, $\omega_{jt}=0(t\neq 0)$ 指第 j 回波没有落入第 t 目标的选通波门。所有可行关联假设事件的构造也正是依据聚矩阵 Ω来产生。可行事件的构造须服从两个条件: (1) 每一回波有且仅有唯一来源 (目标或杂波); (2) 每一目标有且仅有唯一回波。可行关联假设事件的具体意义是从 m_k 个回波中为 T_k 个目标寻找其相应的可行关联数据, 所谓可行数据是指该数据必须落入目标 t 的选通波门内, 而且, 每一数据在同一事件中仅能被选中一次[1] 。由上述两个条件, 目标与回波之间的关联问题实质上是一个组合问题, 从而关联假设事件的数量将随目标数量和回波数量的增大而呈指数增长。形成可行关联事件之后, 可用文献[5]的相应公式计算可行关联事件的条件概率, 其简化形式为[3]

$$P(\epsilon(\hat{\Omega}_i)/Z) = \frac{1}{c}(P_0)^{\min(T_k,m_k)-m_0}\prod_{j:\hat{\omega}_{jt}=1} P_{jt},\quad j=1,2,\cdots,m_k;\quad t=1,2,\cdots,T_k \tag{2}$$

式中, Z 为到当前时刻为止的回波观测全集, c 为归一化常数, m_a 为该关联事件中能够跟回波相关联的目标数, 且

$$P_{jt} = \begin{cases} N(Z_j^t:\ 0, S^t)P_D, & \text{若 } \omega_{jt}=1 \\ 0, & \text{其他} \end{cases} \tag{3}$$

$$P_{0t} = \lambda(1-P_D) = P_0 \tag{4}$$

(3)、(4) 式中, λ为杂波密度, P_D 为目标探测概率, $N(Z_j^t:\ 0, S^t)$ 为具有零均值和协方差矩阵为 S^t的正态分布密度函数。(2) 式中的 c 可以通过和式 $\sum_{\epsilon(\Omega)} P(\epsilon(\hat{\Omega}_i)/Z)$ 求出, 为简便, 下面略去 c 符号。目标 t 跟回波 j 之间的关联后验概率 β_j^t可由依据(2)式的条件概率来计算:

$$\beta_j^t = \sum_{\epsilon(\Omega)} P(\epsilon(\hat{\Omega}_i)/Z)\hat{\omega}_{jt} \tag{5}$$

$$\beta_0^t = 1-\sum_{j=1}^{m_k}\beta_j^t \tag{6}$$

$$j=1,2,\cdots,m_k;\quad t=1,2,\cdots,T_k$$

现在的问题是, 形成了上述的聚矩阵后, 如何来快速搜索和产生关联事件(矩阵)并快速计算后验关联概率。

1.1 分层搜索算法(Layered Searching Algorithm, LSA)

为了说明问题, 我们不妨举一个例子。见图1. 我们可以由图1得出关联聚矩阵(7):

$$\Omega = [\omega_{jt}] = \begin{matrix} \\ 1 \\ 2 \\ 3 \end{matrix} \overbrace{\begin{matrix} 0 & 1 & 2 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}}^{t} \begin{matrix} \\ 1 \\ 2 \\ 3 \end{matrix} \Big\}_j \tag{7}$$

[1] 这种构造关联假设事件的方式是围绕目标进行的, 亦称这种JPDA为目标取向式JPDA. 也可围绕回波而寻找其周围的可行目标, 这种方式称为回波取向式JPDA.

由(7)式用穷举搜索法找到全部的关联矩阵共有8个:

$$\hat{\Omega}_1=\begin{pmatrix}1&0&0\\1&0&0\\1&0&0\end{pmatrix}\quad \hat{\Omega}_2=\begin{pmatrix}0&1&0\\1&0&0\\1&0&0\end{pmatrix}\quad \hat{\Omega}_3=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}\quad \hat{\Omega}_4=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix}$$

$$\hat{\Omega}_5=\begin{pmatrix}1&0&0\\0&1&0\\1&0&0\end{pmatrix}\quad \hat{\Omega}_6=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\quad \hat{\Omega}_7=\begin{pmatrix}1&0&0\\0&0&1\\1&0&0\end{pmatrix}\quad \hat{\Omega}_8=\begin{pmatrix}1&0&0\\1&0&0\\0&0&1\end{pmatrix}$$

我们定义有 L 个目标能跟回波关联的情形为第 L 层。$L=0$ 为第0层, 表示没有一个目标能够跟回波关联。对于当前有 T_k 个目标和 m_k 个回波的情形, 最大层次显然为 $Q_{\max}=\min(m_k, T_k)$。这里假定 $m_k\geq T_k$, $m_k< T_k$ 的情况是对称的。显然, $L=0$ 层只有一个事件, 即所有目标都不能跟回波关联, 如 $\epsilon(\hat{\Omega}_1)$. 注意到, 在 $\hat{\Omega}_1$ 中, 除 $t=0$ 列的元素非0外, 其他元素全为0. 对 $L=1$ 层, 上例中有 $\epsilon(\hat{\Omega}_2)$、$\epsilon(\hat{\Omega}_4)$、$\epsilon(\hat{\Omega}_5)$、$\epsilon(\hat{\Omega}_7)$。又注意到, 在关联矩阵 $\hat{\Omega}_n(n=1,4,6,7)$ 中, 不考虑 $t=0$ 列的元素, 则其他元素中只有一个元素为1. $L=2,3,\cdots,Q_{\max}$ 的情况可依此类推。现在的问题是, 如何按一定次序分层产生关联事件, 以及如何使产生关联事件的次序有利于降低后验概率的计算量。

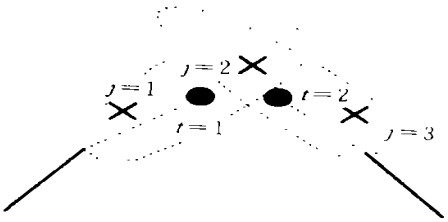

图1 聚矩阵及关联事件形成举例

Zhou[3]和 James[4]的方法是依据聚矩阵的行向量中非0值来进行搜索并以此得出快速算法。我们考察聚矩阵中列向量的情况。对于 $0<L\leq Q_{\max}$ 的层次, 每一个关联矩阵中除 $t=0$ 列外, 其他各列中, 只能在 L 个列中可取其列向量的非0值。对照 $L=1$ 层的情况, 我们可在除 $t=0$ 列外的其他列中的 1 个列向量中取非0值, 亦即, 如果仅仅在第 $t=1$ 列上取非0值, 可得关联事件为 $\epsilon(\hat{\Omega}_2)$、$\epsilon(\hat{\Omega}_3)$; 如果仅仅在第 $t=2$ 列上取非0值, 可得关联事件为 $\epsilon(\hat{\Omega}_3)$、$\epsilon(\hat{\Omega}_6)$, 至此, 已经没有非0值元素可取。这样得出的四个关联事件正好就是 $L=1$ 层的全部可行事件。我们还注意到, 在 $L=1$ 层的搜索处理时, 列序号(即目标序号)的变化范围是从 $t=1$ 到 $t=2$, 即从1到 $Q_{\max}$.

我们再考察 $L=2$ 层的搜索。同样, 我们只考察除 $t=0$ 列外的其他列的列向量中取非0值。图1的例子中, 搜索 $L=2$ 层的事件其意义就是我们可在2个列的列向量中取非0值。本例中就是在第 $t_1=1$ 列和第 $t_2=2$ 列的列向量上取非0值。我们这样搜索, 第一个列在第 $t_1=1$ 列中取列向量的一个非0值后, 按照可行事件形成的两个约束条件, 再在第二个列即在第 $t_2=2$ 列中自上而下取其列向量的一个非0值。如图1例中, 在第 $t_1=1$ 列中自上而下先取 $\omega_{11}=1$ 值后, 在第 $t_2=2$ 列中自上而下分别取 $\omega_{22}=1$ 和 $\omega_{32}=1$, 组合成 $\omega_{11}\omega_{22}$ 和 $\omega_{11}\omega_{32}$, 从而产生两个 $L=2$ 层的关联事件, 即本例中的 $\epsilon(\hat{\Omega}_3)$、$\epsilon(\hat{\Omega}_6)$. 这个处理过程如图2所示。再在 $t_1=1$ 列中自上而下取列向量的第二个非0值, 即 $\omega_{21}=1$, 类似地, 按照可行事件形成的两个约束条件, 在 $t_2=2$ 列中自上而下搜索, 只能取 $\omega_{32}=1$, 组合成 $\omega_{21}\omega_{32}$ 而得一个 $L=2$ 层的关联事件 $\epsilon(\hat{\Omega}_5)$, 至此, 全部层次的关联事件搜索结束。我们将本法的搜索跟 Zhou 和 James 方法比较。按照 Zhou 和 James 的方法对上例的关联事件进行搜索, 搜索得出的关联事件的次序为: $\epsilon(\hat{\Omega}_2)\rightarrow\epsilon(\hat{\Omega}_3)\rightarrow\epsilon(\hat{\Omega}_4)\rightarrow\epsilon(\hat{\Omega}_5)\rightarrow\epsilon(\hat{\Omega}_6)\rightarrow\epsilon(\hat{\Omega}_7)\rightarrow\epsilon(\hat{\Omega}_8)$。用我们的方法, 次序为: $\epsilon(\hat{\Omega}_2)\rightarrow\epsilon(\hat{\Omega}_3)\rightarrow\epsilon(\hat{\Omega}_4)\rightarrow\epsilon(\hat{\Omega}_5)\rightarrow\epsilon(\hat{\Omega}_6)\rightarrow\epsilon(\hat{\Omega}_7)\rightarrow\epsilon(\hat{\Omega}_8)$。

在 $L=2$ 层的搜索中我们注意到, 2个列的列号次序取值没有变化, 即我们搜索所得的组合 $\omega_{1t_1}\omega_{2t_2}$ 中, t_1 取值为从1到1, t_2 取值为从2到2. 回想我们在 $L=1$ 层搜索时的列号取值次序, 情况是, 对于搜索所得的 ω_{1t_1}, t_1 取值为从1到 $Q_{\max}$。还需指出, 在 $L=2$ 层的搜索中, 若 t_1 列和 t_2 列的序号取值保持不变下, 相邻两次搜索的中间结果 $\omega_{1t_1}\omega_{2t_2}$ 之间仅仅有下标 j_2 发生变化。

我们将上述搜索方法推广应用到 $Q_{\max}\geq 3$ 的情况中, 经考察我们可归纳出下列结论:

(1) 设第 L 层的搜索中间结果为 $\omega_{1t_1}\omega_{2t_2}\cdots\omega_{L}\omega_{L}$, 则列号 $t_i(i=1,2,\cdots,L)$ 的取值范围为: $i\leq t_i\leq Q_{\max}-L+i$. 如上例中第一层 t_1 的取值为从1到 $Q_{\max}-1+1=Q_{\max}=2$; 第二层中间结果 $\omega_{1t_1}\omega_{2t_2}$ 中, t_1 取

$$\Omega=[\omega_{jt}]=\begin{pmatrix}\overbrace{0\ \ 1\ \ 2}^{t}\\ 1\ \ 1\ \ 0\\ 1\ \ 1\ \ 1\\ 1\ \ 0\ \ 1\end{pmatrix}\begin{matrix}1\\2\\3\end{matrix}\}j \longrightarrow \hat{\Omega}_2=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix} \qquad \Omega=[\omega_{jt}]=\begin{pmatrix}\overbrace{0\ \ 1\ \ 2}^{t}\\ 1\ \ 1\ \ 0\\ 1\ \ 1\ \ 1\\ 1\ \ 0\ \ 1\end{pmatrix}\begin{matrix}1\\2\\3\end{matrix}\}j \longrightarrow \hat{\Omega}_3=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix}$$

图2 $L=2$层相邻的两次搜索

值为从1到 $Q_{\max}-2+1=1$, t_2取值为从2到 $Q_{\max}-2+2=Q_{\max}=2$.

(2) 在进行第 L 层的搜索时, 若列编号 t_i 不变且跟回波相应的下标 $j_i(i=1,2,\cdots,L-1)$ 也不变时, 则相邻两次搜索的中间结果中仅有最后的元素 $\omega_{L t_L}$ 发生变化, 即从 $\omega_{1t_1}\omega_{2t_2}\cdots\omega_{L-1}\omega_{L-1}\underline{\omega_{L t_L}}$ 变为 $\omega_{1t_1}\omega_{2t_2}\cdots\omega_{L-1}\omega_{L-1}\underline{\omega_{L t_L}'}$, 注意下划线部分的不同。如上例第二层搜索时, $t=1,2$不变下 $\omega_{11}\omega_{22}$ 变为 $\omega_{11}\omega_{32}$.

(3) 上述两结论适用于任何 $L>1$层次的搜索, 我们指出, 由于各层次的搜索是独立进行的并且具有相同逻辑结构, 因而各层次的搜索可递归或并行进行。考虑关联事件产生的两个限制条件, 我们可得出第 L 层的 LSA 搜索伪程序如附录 A.

事实上, 高明的程序员可将上述搜索过程用更加紧凑且可递归调用的形式来实现。由于这一方法具有潜在的并行机制, 也可以通过软件、硬件来分层并行实现。必须指出, 在上述搜索程序中, 需要存储的中间结果仅仅是 $\omega_{1t_1}\omega_{2t_2}\cdots\omega_{L}\omega_{L}$, 另外, 聚矩阵 Ω也需占用一些内存。这一特点跟 Zhou 和 James 的方法不同, 他们的方法中需存储由聚矩阵 Ω产生的各行向量及其调用情况。对一个系统资源有限的传感器如雷达、红外跟踪搜索仪而言, 节省内存、软件紧凑、程序的可递归性以及可并发实现, 都正是工程设计人员所想要达到的目标。

1.2 后验概率 β_j^i 的快速计算

后验概率 β_j^i 的快速计算将基于上述搜索过程的结论2进行。后验概率 β_j^i 的计算是从式(2)的 $P(\epsilon(\hat{\Omega})/Z)$ 计算开始的。在第 L 层, 式(2)可重写为

$$P(\epsilon(\hat{\Omega})/Z)=\frac{1}{c}(P_0)^{Q_{\max}-L}\prod_{j\in\{j_t\}=1}P_{jt}.\quad j=j_1,j_2,\cdots,j_L;\ t=t_1,t_2,\cdots,t_L \tag{8}$$

结合我们的关联事件产生方式, 则对于同一 L 层, 因子 $(P_0)^{Q_{\max}-L}$ 仅仅需要计算一次。L 阶连乘因子 $\prod P_{jt}$ 的计算是 JPDA 方法中很费时的, 但在本法中, 该因子是依据搜索中间结果 $\omega_{1t_1}\omega_{2t_2}\cdots\omega_L\omega_L$ 进行的, 而且, 在计算形成聚矩阵时, 跟聚矩阵中非0元素一一对应的正态概率分布函数 P_{jt} 也同时计算好, 存储起来备后面的计算使用。由结论2, 如果列编号 $t_i(i=1,2,\cdots,L)$ 保持不变, 我们则将跟 $\omega_{1t_1}\omega_{2t_2}\cdots\omega_{L-1}\omega_{L-1}$ 相应的 $L-1$阶连乘因子 $\prod_{j\in\{j_t\}=1}P_{jt}(j=j_1,j_2,\cdots,j_{L-1};\ t=t_1,t_2,\cdots,t_{L-1})$ 的计算结果暂时存储起来, 本层的下一个关联事件产生后, 只需将跟 $\omega_{L t_L}$ 相应的 $P_{j_L t_L}$ 值跟暂存的 $L-1$阶连乘因子 $\prod_{j\in\{j_t\}=1}P_{jt}(j=j_1,j_2,\cdots,j_{L-1};\ t=t_1,t_2,\cdots,t_{L-1})$ 相乘即可; 如果列编号 t_i 有变化, 则 L 阶连乘因子 $\prod_{j\in\{j_t\}=1}P_{jt}(j=j_1,j_2,\cdots,j_L;\ t=t_1,t_2,\cdots,t_L)$ 中所有 L 个因子都需要更新且需计算一次连乘, 共需作 $L-1$次乘法, 用所得的新的 $L-1$阶连乘因子 $\prod_{j\in\{j_t\}=1}P_{jt}(j=j_1,j_2,\cdots,j_{L-1};\ t=t_1,t_2,\cdots,t_{L-1})$ 替代暂存在内存中的原 $L-1$阶连乘因子。这一计算过程是结合关联事件的搜索进行的, 在上述伪程序中的计算 $P(\epsilon(\hat{\Omega})/Z)$ 部分执行。由于在所产生的两个相邻事件之间的计算结果有继承性, 因而降低了计算量, 后面我们将进行定量分析。所有的 $P(\epsilon(\hat{\Omega})/Z)$ 计算完后, 可依据(5)、(6)式计算所有的后验概率 β_j^i.

1.3 算法的实现

由上面的讨论可知, 我们可以用递归调用方法或者分层并行计算的方法来实现快速 JPDA 算法。两种方法的原理见图3和图4.

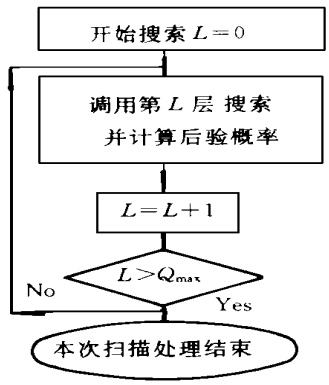

图3 递归式实现

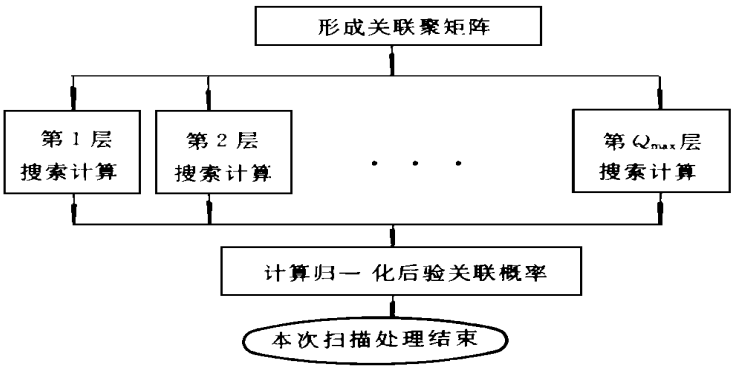

图4 并行方式实现

1.4 计算量的定量分析

我们来分析本法所需的计算量, 这里只给出求 β_i^j 所需计算 P_{ji} 的次数, 未计所节约的加法次数, 这部分节约的加法计算量也是非常可观的。我们以极端的例子来讨论, 即聚矩阵 Ω 为全1元素矩阵。从伪程序可知, 在 L 层搜索中, 关于 t_i 有 L 层循环, 每一循环要执行 $Q_{\max}-L+1$ 次; 关于 j_i 也有 L 层循环, 第一层循环可有 m_k 种选值, 第二层循环可有 m_k-1 种选值, 并考虑上述的计算继承性, 这样, 第 L 层所需的乘法次数为

$$N_L = C_{Q_{\max}}^{L}[m_k(m_k-1)\cdots(m_k-L+2)\quad(L-1)+m_k(m_k-1)\cdots(m_k-L+2)\quad(m_k-L)]$$

$$= C_{Q_{\max}}^{L}(m_k-1)\frac{m_k!}{(m_k-L+1)!} \tag{9}$$

则计算 β_i^j 所需的乘法次数为各层乘法数目之和, 即

$$N = \sum_{L=2}^{Q_{\max}} N_L = \sum_{L=2}^{Q_{\max}} C_{Q_{\max}}^{L}(m_k-1)\frac{m_k!}{(m_k-L+1)} \tag{10}$$

我们将本文的 LSA 方法跟 Zhou[3]的 DFS 方法就所需的乘法次数进行比较, 见表1.

表1 算法所需乘法次数的比较, DFS 的乘法数 / LSA 的乘法数

T_k	m_k	DFS/ LSA
3	6	318/240
4	8	5072/4258
5	10	96890/86764
6	12	2218992/2078347

1.5 平均计算量的定量分析

考虑通常情况下在每一目标的预测波门内有2～3个回波, (这个假设相对合乎实际情况), 则观测回波的总数一般都落入下面的闭区间内, 即

$$T_k \le m_k \le 2T_k+1 \tag{11}$$

由这一假设出发, Zhou 方法的计算量中的乘法次数上限为

$$M(T_k, m_k) < 2^* 4^{T_k} - 2 - 3T_k - 3^{T_k} \tag{12}$$

本文中 LSA 方法的第 L 层乘法次数为

$$N_L = C_{Q_{\max}}^{L}\overbrace{3\ \ 3\cdots 3}^{L-1}(L-1) + C_{Q_{\max}}^{L}\overbrace{3\ \ 3\cdots 3}^{L-1}(3-1) \tag{13}$$

$$= C_{Q_{\max}}^{L}3^{L-1}(L+1)$$

则本文 LSA 方法的计算量中的乘法次数上限为

$$N = \sum_{L=2}^{Q_{\max}} N_L = \sum_{L=2}^{Q_{\max}} C_{Q_{\max}}^{L}3^{L-1}(L+1) \tag{14}$$

我们将本文 LSA 方法跟 Zhou 的 DFS 方法就所需的乘法次数上限进行比较, 见表2.

表2 算法所需乘法次数上限的比较

T_k	DSF	LSA
3	90	63
4	417	333
5	1788	1611
6	7443	7497
7	30558	34119

注意到当 $T_k \geq 7$时, 本文的 LSA 方法的乘法数上限比 DFS 方法的上限略大一些。但是, $N_L = C_{Q_{\max}}^{L} 3^{L-1}(L+1)$ 是第 L 层乘法次数的稀疏上限, 实际情况比这一数值要小得多, 因为我们在推导(13) 式时的条件放得比 DFS 还松。我们也将本文LSA 方法的乘法数上限跟 James 方法的上限作比较, 见表3.

表3 James 的快速 JPDA 方法跟本文 LSA 方法所用乘法数目的比较, JPDA / LSA

回波数目	目标数目 T_k				
	3	4	5	6	7
T_k	51/30	260/131	1045/542	3654/2496	11655/8637
T_k+1	78/37	432/178	1830/795	6624/3450	21603/11601
T_k+2	111/43	672/225	3030/911	11466/3615	38661/14841
T_k+3	150/51	992/245	4780/1027	19020/4365	66626/16695
T_k+4	195/55	1404/265	7235/1143	30360/4653	110922/18549
T_k+5	–	1920/285	10570/1235	46824/5023	178836/20403
T_k+6	–	–	14980/1327	70044/5481	279058/22095
T_k+7	–	–	–	101976/5897	426636/23787
T_k+8	–	–	–	–	634473/25479

由上面的分析可知, 本文的 LSA 方法具有快速、可递归、省内存、可并行分层实现的特点, 这些特点为本文方法的实际应用提供了基础。限于篇幅, JPDA 算法的其他部分的内容如归一化常数的计算、JPDA 滤波方程组等, 不再一一列举, 读者可从参考文献查到。

2 算法的计算机仿真实验

实验采用的目标动态模型及测量模型如下面方程组所示。假定采样周期为1s, 监控的区域为40km ×40km 的范围, 设置10个目标的初始位置和速度如表4所示(同文献[3])。

$$\boldsymbol{x}_{k+1} = \Phi_k \boldsymbol{x}_k + \boldsymbol{G}\boldsymbol{w}_k \tag{15}$$

$$\boldsymbol{Z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \nu_k \tag{16}$$

式中, $\boldsymbol{x}_k = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k]^{\mathrm{T}}$, $\boldsymbol{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $\boldsymbol{G} = \begin{bmatrix} T^2/2 & T & 0 & 0 \\ 0 & 0 & T^2/2 & T \end{bmatrix}^{\mathrm{T}}$, $\Phi_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

$E\{\nu_k\} = 0$, $E = \{\nu_k \nu_j^{\mathrm{T}}\} = \boldsymbol{R}_k \delta(k-j)$, $\boldsymbol{R}_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_x^2 \end{bmatrix}$, $\sigma_x = \sigma_y = 0.15\mathrm{km}$, $\boldsymbol{w}_k = [w_1(k) \quad w_2(k)]^{\mathrm{T}}$, $E\{\boldsymbol{w}_k\} = 0$, $E\{\boldsymbol{w}_k \boldsymbol{x}_j^{\mathrm{T}}\} = q\delta(k-j)$, 一般取 $q = 10^{-4}$. δ_{kj}为 Kronecker 标记符。

设传感器对目标的检测概率为 $P_D = 0.9$, 杂波点的数目服从 Poisson 分布, 杂波点的产生将按照参数为 $\lambda = 2.0$的 Poisson 分布函数产生(每一目标的预测波门中平均有0.5～3个杂波)。关联波门门限 G^2

选为17.

我们设置两个实验题目, 一是考察 LSA 方法在一次样本计算中所用的平均时间, 二是考察该算法的平均成功率(即每一百次实验中不丢失目标的正确跟踪次数)。实验在 PC586-133上进行, 我们采用上面提到的递归方式来做实验。实验结果见表5及表6, 图5～图8分为对5个、10个目标正确跟踪(60次扫描)的例子。

3 结论

我们讨论快速递归计算 JPDA 的 LSA 算法, 包括关联假设事件的分层搜索产生方法, 后验概率的快速计算, 分析了本文方法的计算量。本文的 LSA 方法具有快速、省内存、可递归或并行实现的特点, 这些特点为本文方法的实际应用提供了基础。为了证实该方法的有效性, 做了 Monte Carlo 试验。本文方法特别方便于在工程实践中用硬件来并行实现。另外, 由于本文方法的计算量适中, 并且有着比次优 JPDA 算法更加优良的性能, 而算法仅仅比次优 JPDA 算法稍微复杂一点, 因而, 本文的方法是一种比较有效的快速 JPDA 算法。

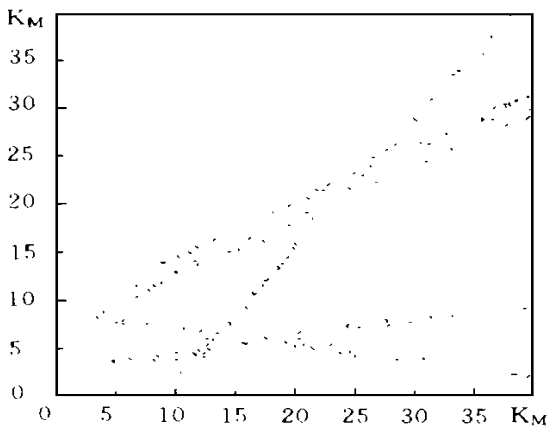

图5 在预测波门内得到的五个目标的观测报告(含目标数据和杂波)

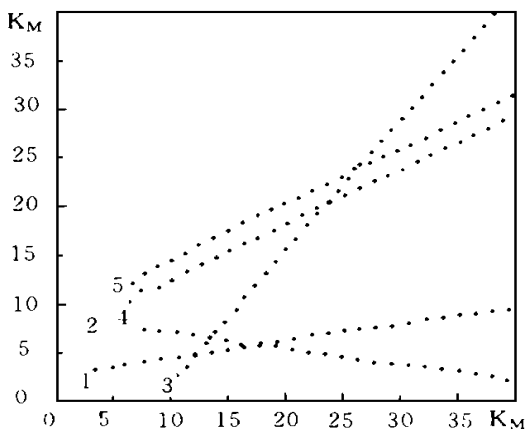

图6 图五的五个目标的跟踪结果

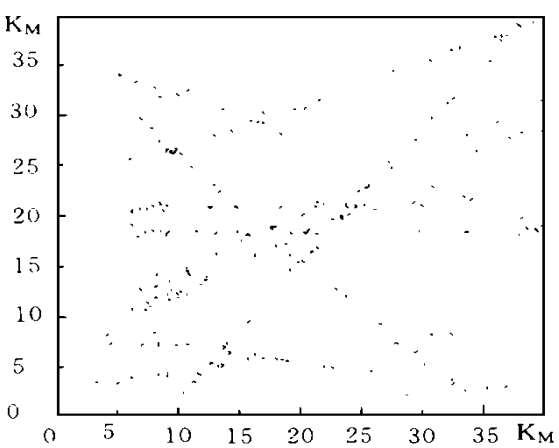

图7 在预测波门内得到的十个目标的观测报告(含目标数据和杂波)

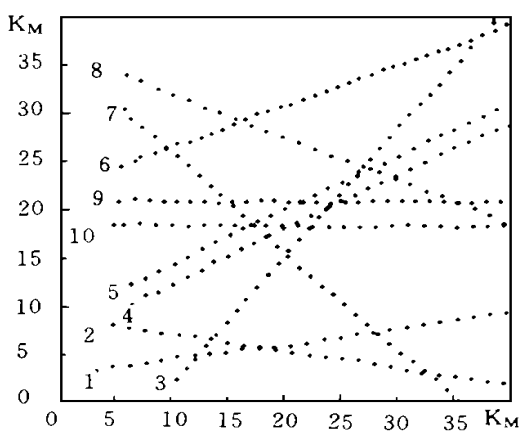

图8 图八的十个目标的跟踪结果

表4 10个目标的初始位置和速度

目 标	初始位置(km)		速度(km/s)	
t	x	y	$\dot{x}$	$\dot{y}$
1	3.0	3.0	0.72	0.12
2	3.0	7.8	0.72	-0.12
3	10.0	2.0	0.45	0.60
4	6.0	10.0	0.62	0.35
5	6.0	12.0	0.62	0.35
6	5.0	24.0	0.50	0.22
7	5.0	30.0	0.50	-0.50
8	4.0	34.0	0.70	-0.31
9	5.0	20.5	0.62	0.00
10	4.0	18.0	0.70	0.00

表5 LSA 法和直接计算法数据关联和目标跟踪所用的 CPU 时间

	目标数目	回波数目	采样次数	平均 CPU 时间/s
LSA 法	5	11	16	0.055
	6	12	21	0.105
	7	15	18	0.314
	8	18	15	0.771
	9	18	15	2.001
	10	21	8	16.513
直接计算法	5	11	16	0.775
	6	12	21	2.189
	7	15	18	5.585
	8	18	15	21.821
	9	18	12	48.333
	10	21	8	456.383

表6 跟踪实验的成功率(扫描次数为60)

目标数目	实验重复次数	成功次数	成功率%
5	100	100	100
6	100	100	100
7	100	100	100
8	100	99	99
9	80	76	95
10	80	71	88.75

参考文献

1 Fitzgerald R J. Development of practical PDA logic for tracking by microprocessor. Proceedings of American Controls Conference, Seattle, WA, June 1986: 889～898

2 Roecker J A. A class of near optimal JPDA algorithms. IEEE Trans. AES. 1994, 30(2): 504～510

3 Zhou B, Bose N K. Multitarget tracking in clutter: Fast algorithm for data association. IEEE Trans. AES- 29, 1993, (2): 352～363

4 Fisher L J, Casasent D P. Fast JPDA multitarget tracking algorithm. Applied Optics, 1989, 28(2): 371～376

5 Bar-shalom Y, Blackman T E. Tracking and Data Association. New York: Academic Press, 1988

6 Blackman S S. Multiple Target Tracking with Radar Applications, Delham, MA: Artech House, 1986

7 Houles A, Bar-shalom Y. Multisensor tracking of a maneuvering target in clutter. IEEE Trans. AES- 25, 1989, (2): 176～189

附录 A: LSA 算法伪程序

For t_1 = 1 to Q_{max} - L + 1
  For t_2 = 2 to Q_{max} - L + 2
  ⋮
  For t_L = L to Q_{max} - L + L
} 共 L 级循环

Begin:
  For j_1 = 1 to m_k

    Begin: if $\omega_{j_1 t_1} \neq 0$ continue;
  for j_2 = 1 to m_k
    Begin: if $j_2 = j_1$ continue;
          if $\omega_{j_2 t_2} = 0$ continue;

  for j_{L-1} = 1 to mk
    if $j_{L-1} = j_{L-2}, j_{L-3}, \cdots, j_2, j_1$. continue;
      计算并存储 Buffer = $\prod_{t=1}^{L-1} P_{j_t}(j = j_1, \cdots, j_{L-1};\ t = t_1, \cdots, t_{L-1})$
      for j_L = 1 to mk
        Begin:
          if $j_L = j_{L-1}, j_{L-2}, \cdots, j_2, j_1$ continue;
          if $\omega_{j_L t_L} = 0$ continue;
            输出关联事件并计算式(3)的 $p(\epsilon(\hat{\Omega}/Z) =$ Buffer $P_{j_{L-1} t_L}$
              计算后验关联概率(未归一化)
          end
      end

end (共2×L 级)