doi:10.11887/j.cn.201904022 http://journal.nudt.edu.cn

相似分布特性准则下的高斯混合项聚类 - 合并算法*

徐 洋,方洋旺,伍友利,张丹旭
(空军工程大学 航空工程学院, 陕西 西安 710038)

摘 要:针对高斯混合模型估计非高斯系统时高斯混合项呈指数级增长问题,提出一种基于相似分布特性准则的聚类 - 合并方法。通过分析高斯混合项的分布特性,基于扩展积分均方误差代价函数搜索最优置信范围,并对混合项进行高斯聚类,进而获得具有不同分布特性的高斯簇。为防止高斯簇间对高斯子项的重复利用,引入局部最近邻思想对交叉高斯项进行重新分配。采用并行多元素合并方法对高斯簇中的混合项进行合并,在保证无偏性基础上减少下一时刻混合项数量。仿真结果表明,改进算法在保证跟踪精度的同时还可有效提高算法效率。

关键词:高斯混合模型;相似分布准则;高斯聚类;扩展积分均方误差代价函数;多元素融合

中图分类号:TN391 **文献标志码**:A **文章编号**:1001-2486(2019)04-156-09

Cluster-merge method for the Gaussian mixture components based on the similarity distribution criterion

XU Yang, *FANG Yangwang*, *WU Youli*, *ZHANG Danxu*
(Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China)

Abstract: To solve the problem of the exponential growth of the Gaussian mixture components for estimating the state of the non-Gaussian system with the Gaussian mixture model, a cluster-fusion algorithm based on the similarity distribution criterion was proposed. According to that criterion, the Gaussian components were then clustered into different Gauss clusters based on the optimal confidence interval, derived by minimizing the extended integral square error cost function. Meanwhile, to avoid the reuse of the cross components, the local nearest neighbor approach was introduced to re-allocate these cross ones. Then, the components in the clusters were merged by the multi-element mergence method to keep with the unbiased property, which can decrease the number of the mixture components sharply. The results show that the proposed algorithm can not only reduce the running time, but also guarantee the tracking performance with a proper confidence interval.

Keywords: Gaussian mixture model; similarity distribution criterion; Gauss cluster; extended integral square error cost function; multi-element fusion

在目标跟踪领域中,系统模型误差不可避免,产生的原因主要有目标运动状态的变化及目标运动模型的“双非”(非线性、非高斯性)问题[1]。解决上述问题的主要方法有:提高运动模型自适应能力,如自适应机动模型[2];改进滤波算法性能,如粒子滤波器(Particle Filter, PF)[3];增强跟踪模型的适用范围以解决“双非”问题,如高斯混合模型(Gaussian Mixture Model, GMM)[4]。由于自适应机动模型人为设定因素过多且适用性还有待检验,而PF算法计算复杂度过高,难以应用于战时平台,为此传统方法是引入GMM对“双非”系统下的目标状态进行估计。

GMM是单一高斯概率密度函数(Probability Density Function, PDF)的扩展,理论上它可以平滑地近似任意非线性函数[5],但其本身仍存在不足:当动态方程的噪声项及初始状态中不止一个采用GMM表示时,算法的计算复杂度会呈指数级增长[6]。为此学者们分别从不同角度提出解决方案,这些方法大体可概括为以下两种:基于剪枝的删减策略与基于聚类的合并策略。

基于剪枝的删减策略即是基于某种测量准则剔除影响程度较低的混合项。其中最为简单的一种即为删掉具有较低权重的高斯混合项[7],但该方法对于具有多峰分布的PDF近似效果并不佳[8]。为此文献[9]提出了一种基于KL(Kullback-Leibler)散度的剪枝方法。该方法在

* **收稿日期**:2018-03-22
基金项目:航空科学基金资助项目(20175596020)
作者简介:徐洋(1989—),男,吉林吉林人,博士研究生,E-mail:mingze0108@126.com;
方洋旺(通信作者),男,教授,博士,博士生导师,E-mail:ywfang2008@sohu.com

剪枝的同时依然可保留原分布峰值特征,避免了上述问题的发生。文献[6]更是直接利用初始状态筛选有效的噪声混合项,从源头降低了混合项数量。

基于聚类的合并策略即是基于某种距离标准将相似的混合项加以合并,该削减策略得到了广泛认可。文献[10]通过寻找混合项中两两之间具有最小 KL 散度的混合项,然后将其合并达到削减数量的目的。但该方法每合并一次都要重新搜索所有混合项,时效性较差。文献[11]在此基础上,又利用了聚类合并的思想,虽提高了精度,但算法复杂度仍很高。文献[12]更是对典型合并方法进行了系统性分析与比较。正是以上 GMM 削减方法的提出,使得 GMM 得以应用于众多领域[13-17]。

为了充分利用混合项自身的分布信息,简化整体运算流程,在保证精度的同时提高算法效率,本文提出了一种复杂度适中、近似效果较好的高斯聚类 - 合并方法。首先基于高斯混合项的后验分布特性及置信区间,建立相似分布特性准则,并采用扩展积分均方误差(Extended Integral Square Error,EISE)准则求解最优置信区间,完成对混合项的高斯聚类;然后针对各高斯簇间存在的交叉项复用问题,利用局部最近邻(Local Nearest Neighbor, LNN)思想将其重新分配;最后在各高斯簇中利用多元合并方法,减少下次迭代混合项数量。通过仿真分析了有效置信范围的可行域,缩窄了置信范围的搜索空间,并通过两个典型场景与基于数量限制的高斯混合项[7](Gaussian Mixture Reduction based on Pruning, GMRP)、Runnall[10]及基于聚类的高斯混合项削减方法[11](Gaussian Mixture Reduction based on Clustering, GMRC)进行了对比,验证了本文算法具有较高的效费比。

1　GMM 及改进方法

设线性系统的状态方程及观测方程为

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{u}_k \\ z_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \end{cases} \tag{1}$$

式中:$k \leqslant N_K$ 为时间序列,N_K 为序列长度;状态量 $\boldsymbol{x}_k \in \mathbf{R}^{m_x}$,其中 m_x 为 $\boldsymbol{x}$ 的维度,初始状态分布为 $p(\boldsymbol{x}_0)$,转移概率 $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ 满足马尔科夫过程。观测量 $z_k \in \mathbf{R}^{m_z}$,其中 m_z 为观测量 z 的维度,$p(z_k|\boldsymbol{x}_k)$ 为系统似然函数;k 时刻状态量及观测量的集合分别为 $\boldsymbol{x}_{0:k} = \{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_k\}$ 及 $\boldsymbol{Z}_k = \{z_0, \cdots, z_k\}$;$\boldsymbol{u}_k$ 及 $\boldsymbol{v}_k$ 分别为状态噪声和观测噪声;$\boldsymbol{F}$ 为状态转移矩阵,$\boldsymbol{H}$ 为观测矩阵。

利用 GMM 表示的总体概率密度为

$$\begin{cases} f(\boldsymbol{x}|\Omega^N) \triangleq \sum_{i=1}^{N} \omega_i \cdot \mathrm{N}(\boldsymbol{x};\boldsymbol{\mu}^{(i)}, \boldsymbol{P}^{(i)}) \\ \text{s.t.} \ \sum_{i=1}^{N} \omega_i = 1, \ \forall \omega_i \geqslant 0 \end{cases} \tag{2}$$

式中:$\mathrm{N}(\boldsymbol{x};\boldsymbol{\mu}^{(i)}, \boldsymbol{P}^{(i)})$ 为多维高斯分布的 PDF;N 为 GMM 中混合项数量;Ω^N 代表均值、方差及权重的集合;ω_i 代表第 i 个正态分布的权值。将初始状态、状态噪声及测量噪声中的两个或三个用 GMM 加以表示,会导致在状态更新及时间更新阶段,系统的高斯混合项呈现指数级增长。针对该点不足,在综合考虑高斯混合项权重信息和分布特性的基础上,本文提出如下基于相似分布特性准则的聚类 - 合并方法以动态削减高斯混合项。

2　混合项聚类 - 合并算法

2.1　高斯聚类

服从高斯分布的混合项之间无法定义为完全独立,而且聚类问题本身在统计上也是无法识别的[18],因此每种聚类方法都包含着主观的聚类概念。本文所提的高斯聚类方法是以混合项具有较小方差[18]为前提,依据相似分布特性准则加以聚类。若混合项中存有较大方差项,可通过设置阈值 δ 将其排除在聚类项之外。其中对于聚类依据——相似分布特性准则的定义如下:若 p 个具有较小方差的混合项的分布中心相互处于对方的 $\boldsymbol{P}^*$ 置信范围内,则认为这些混合项在分布上具有 $\boldsymbol{P}^*$ 相似特性,从而将其归为 $\boldsymbol{P}^*$ 置信范围下的同一高斯簇。该准则的现实意义即是:若高斯混合项之间分布形状足够相似、分布中心足够接近,则其合并后的 PDF 将服从单峰分布,且总体分布特性服从类高斯分布[19]。为了搜寻可同属一个高斯簇的混合项,定义 k 时刻高斯混合项的分布中心向量 $\boldsymbol{\Pi}$ 为

$$\boldsymbol{\Pi}_k = [\boldsymbol{u}_k^1, \quad \boldsymbol{u}_k^i, \quad \cdots, \quad \boldsymbol{u}_k^N] \tag{3}$$

其中,$\boldsymbol{u}_k^i$($i = 1, 2, \cdots, N$)为第 i 个高斯混合项的分布中心。因 GMM 中所有混合项都为高斯分布,故此处 $\boldsymbol{u}_k^i$ 等价于混合项的一阶原点矩。

定义如式(4)所示判别矩阵 $\boldsymbol{R}$ 用以判断混合项之间的相似性。

$$\boldsymbol{R}_k = \begin{bmatrix} r_k^{11} & r_k^{12} & \cdots & r_k^{1N} \\ 0 & r_k^{22} & \cdots & r_k^{2N} \\ \vdots & \vdots & r_k^{ij} & \vdots \\ 0 & 0 & \cdots & r_k^{NN} \end{bmatrix} \tag{4}$$

其中，

$$r_k^{ij}=\begin{cases}1 & g_{P^*}^j(\boldsymbol{u}_k^i)\leqslant 0,g_{P^*}^i(\boldsymbol{u}_k^j)\leqslant 0\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

$g_{P^*}^i(\cdot)$表示混合项 i 置信范围为 $\boldsymbol{P}^*$ 的空间曲面在水平面上的投影方程，若 $g_{P^*}(\boldsymbol{x})\leqslant 0$，则表示 $\boldsymbol{x}$ 位于该置信范围内，反之则不在。$r_k^{ij}(i=1,\cdots,N;j=i,\cdots,N)$ 表示高斯混合项 i 和 j 间的分布相似标志，若判定为相似，$r_k^{ij}=1$，则基于 $\boldsymbol{P}^*$ 置信范围两者归于同一高斯簇；若不相似，$r_k^{ij}=0$。

为了快速完成聚类，首先按行搜索判别矩阵，并对每个混合项附上隶属簇标签

$$\Omega_k^i=\{\boldsymbol{c}_k^i(f_i),\ \boldsymbol{c}_k^j(f_i)\mid r_k^{ij}=1\} \tag{6}$$

式中：Ω_k^i 表示第 i 个高斯簇；$\boldsymbol{c}_k^j(f_i)$ 表示第 j 个混合项属于第 i 个高斯簇。为方便标记，后文式子均省略标签项 f。避免混合项的重复利用，此处限制每个混合项能且仅能隶属于一个高斯簇。对于存在交叉项的高斯簇之间需对交叉项进行二次分配，步骤如下：

步骤 1：将交叉的高斯簇分成以下两种

$$\Omega_k^i\cap\Omega_k^j=\begin{cases}H_1:\ \{\boldsymbol{c}_k^m,\cdots\}\\ H_2:\ \Omega_k^j\end{cases} \tag{7}$$

式中：H_1 表示两个高斯簇存在交叉项；H_2 表示 $\Omega_k^j\subseteq\Omega_k^i$。

步骤 2：重新分配交叉项。

1）H_1：将交叉项依 LNN 思想重新分配。其过程如下：

①将交叉项 $\boldsymbol{c}_k^m$ 从两个高斯簇中删除

$$\begin{cases}\Omega_k^i=\Omega_k^i\setminus\{\boldsymbol{c}_k^m,\cdots\}\\ \Omega_k^j=\Omega_k^j\setminus\{\boldsymbol{c}_k^m,\cdots\}\end{cases} \tag{8}$$

②计算交叉项与所属高斯簇间距离

$$d_k^{im}=\min[\,|\boldsymbol{u}_k^{i_1}-\boldsymbol{u}_k^m|,\cdots,|\boldsymbol{u}_k^{i_t}-\boldsymbol{u}_k^m|\,] \tag{9}$$

式中，$\boldsymbol{u}_k^{i_t}(t=1,\cdots,N_k^i)$ 为高斯簇 i 中的第 t 个高斯混合项，N_k^i 为高斯簇 i 中高斯混合项的个数。

③判断所属高斯簇。若 $d_k^{im}\leqslant d_k^{jm}$，则 $\boldsymbol{c}_k^m\in\Omega_k^i$，反之 $\boldsymbol{c}_k^m\in\Omega_k^j$。

④重复步骤①～③直至交叉项分配完。

2）H_2：保留 Ω_k^i 和它的补集 $(\Omega_k^j)^*$。

由上述可知：$\boldsymbol{P}^*$ 对于聚类结果影响很大。若 $\boldsymbol{P}^*$ 过大，则可能使得不具有相似分布特性的混合项归并为同一簇，使得混合项发生多样性退化，最终导致近似多峰分布时效果变差；若 $\boldsymbol{P}^*$ 过小，则会使得聚类过细，无法明显减少高斯混合项个数而使得实效性达不到要求。因此合理地选择 $\boldsymbol{P}^*$ 对于提高算法的效费比尤为重要。为此，本文提出基于 EISE 代价函数求解 $\boldsymbol{P}^*$。

2.2 基于 EISE 函数的置信范围搜寻算法

为了筛选合适的 $\boldsymbol{P}^*$，本文受文献[20]中所提 ISE 函数启发，通过增加正则化项，在限定高斯簇个数的同时使得近似误差逐渐降低，得式(10)所示 EISE 函数。

$$\min_{P^*}J_{\mathrm{S}}=\sum_{i=1}^{N_k^h}\int[f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*}-q(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})]^2\mathrm{d}\boldsymbol{c}_k+\alpha\lg N_k^h \tag{10}$$

其中：$f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*}$ 表示在当前置信范围 $\boldsymbol{P}^*$ 下高斯簇 i 内的估计 PDF；$q\{\cdot\}$ 表示混合项的合并函数，$q(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})$ 表示在置信范围 $\boldsymbol{P}^*$ 下高斯簇采用多元素合并方法获得的合并项；N_k^h 表示高斯簇个数；α 是正则化常数，用于对近似精度及时效性进行折中。因 $f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*}$ 及 $q(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})$ 都为高斯分布，故式(10)中积分项可简化为

$$\int[f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*}-q(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})]^2\mathrm{d}\boldsymbol{c}_k=J_{\mathrm{hh}}-2J_{\mathrm{hr}}+J_{\mathrm{rr}} \tag{11}$$

其中，

$$J_{\mathrm{hh}}=\int(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})^2\mathrm{d}\boldsymbol{c}_k$$

$$J_{\mathrm{hr}}=\int(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})q(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})\mathrm{d}\boldsymbol{c}_k$$

$$J_{\mathrm{rr}}=\int[q(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})]^2\mathrm{d}\boldsymbol{c}_k$$

式(11)两个高斯 PDF 可表示为

$$\begin{cases}f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*}=\sum_{j=1}^{N_i}p_j\cdot\mathrm{N}\{\boldsymbol{c}_k;\boldsymbol{u}_j,\boldsymbol{P}_j\}\\ q(f\{\boldsymbol{c}_k|\Omega_k^i\}_{P^*})=\bar{p}\cdot\mathrm{N}\{\boldsymbol{c}_k;\bar{\boldsymbol{u}},\bar{\boldsymbol{P}}\}\end{cases} \tag{12}$$

其中：$\{p_j,\boldsymbol{u}_j,\boldsymbol{P}_j\}$ 表示高斯簇内混合项的权值、均值和方差；$\{\bar{p},\bar{\boldsymbol{u}},\bar{\boldsymbol{P}}\}$ 表示采用合并算法后高斯项的权重、均值和方差；N_i 为高斯簇 i 内混合项数量。

将式(12)代入到式(11)中，可得

$$\begin{cases}J_{\mathrm{hh}}=\sum_{j=1}^{N_i}\sum_{t=1}^{N_i}p_jp_k\int\mathrm{N}\{\boldsymbol{c}_k;\boldsymbol{u}_j,\boldsymbol{P}_j\}\cdot\mathrm{N}\{\boldsymbol{c}_k;\boldsymbol{u}_t,\boldsymbol{P}_t\}\mathrm{d}\boldsymbol{c}_k\\ J_{\mathrm{hr}}=\sum_{j=1}^{N_i}p_j\bar{p}\int\mathrm{N}\{\boldsymbol{c}_k;\boldsymbol{u}_j,\boldsymbol{P}_j\}\cdot\mathrm{N}\{\boldsymbol{c}_k;\bar{\boldsymbol{u}},\bar{\boldsymbol{P}}\}\mathrm{d}\boldsymbol{c}_k\\ J_{\mathrm{rr}}=\bar{p}^2\int\mathrm{N}\{\boldsymbol{c}_k;\bar{\boldsymbol{u}},\bar{\boldsymbol{P}}\}\cdot\mathrm{N}\{\boldsymbol{c}_k;\bar{\boldsymbol{u}},\bar{\boldsymbol{P}}\}\mathrm{d}\boldsymbol{c}_k\end{cases} \tag{13}$$

两个高斯 PDF 的乘积可以简化为

$$\mathrm{N}\{\boldsymbol{x};\boldsymbol{u}_1,\boldsymbol{P}_1\}\cdot\mathrm{N}\{\boldsymbol{x};\boldsymbol{u}_2,\boldsymbol{P}_2\}=\alpha\mathrm{N}\{\boldsymbol{x};\boldsymbol{u}_3,\boldsymbol{P}_3\} \tag{14}$$

其中，

$$\alpha=\mathrm{N}\{\boldsymbol{u}_1;\boldsymbol{u}_2,\boldsymbol{P}_1+\boldsymbol{P}_2\}$$

$$\boldsymbol{P}_3 = (\boldsymbol{P}_1^{-1} + \boldsymbol{P}_2^{-1})^{-1}$$

$$\boldsymbol{u}_3 = \boldsymbol{P}_3(\boldsymbol{P}_1^{-1}\boldsymbol{u}_1 + \boldsymbol{P}_2^{-1}\boldsymbol{u}_2)$$

将式(14)代入式(13)，又因高斯子项的积分是在整个输入空间，故其积分结果为 1，则最后只剩 α。则式(13)可表示为

$$\begin{cases} J_{\mathrm{hh}} = \sum_{j=1}^{N_i}\sum_{t=1}^{N_i} p_j p_t \cdot \mathrm{N}\{\boldsymbol{u}_j;\boldsymbol{u}_t,\boldsymbol{P}_t + \boldsymbol{P}_j\} \\ J_{\mathrm{hr}} = \sum_{j=1}^{N_i} p_i \bar{p} \cdot \mathrm{N}\{\boldsymbol{u}_j;\bar{\boldsymbol{u}},\boldsymbol{P}_j + \bar{\boldsymbol{P}}\} \\ J_{\mathrm{rr}} = \bar{p}^2 \cdot \mathrm{N}\{\bar{\boldsymbol{u}};\bar{\boldsymbol{u}},2\bar{\boldsymbol{P}}\} \end{cases} \tag{15}$$

$$\text{s.t.} \quad \sum_{j=1}^{N_i} p_j = 1, \forall p_j \geqslant 0$$

至此，代价函数等于高斯簇中混合项之间的乘积加上每一个混合项与合并后项的乘积，加上合并后混合项的概率平方，再加上一个关于高斯簇数量的惩罚函数。因高斯簇已将全部混合项划分为多个高斯簇，而高斯簇内混合项数量也随着 $\boldsymbol{P}^*$ 而变化——$\boldsymbol{P}^*$ 越小，混合项数量越少，这对于算法效率的提高是有帮助的，但同时也使得惩罚项变大。为了更有效地寻找最优解，可先缩小搜索域后利用 MATLAB 自带函数 fmincon 求解。

2.3 无偏多元素合并算法

为了降低聚类复杂度，在获得混合项后可剔除掉连续两步权值都非常小的混合项，此处可设阈值为 0.01。之后基于文献[20]提出的合并方法，在保证系统状态无偏的前提下，对各高斯簇内的混合项进行合并

$$\alpha_k^{\mathrm{merged}} = \sum_{j=1}^{N_i} \alpha_k^j \tag{16}$$

$$\boldsymbol{u}_k^{\mathrm{merged}} = \frac{1}{\alpha_k^{\mathrm{merged}}}\sum_{j=1}^{N_i} \alpha_k^j \boldsymbol{u}_k^j \tag{17}$$

$$\boldsymbol{P}_k^{\mathrm{merged}} = \sum_{j=1}^{N_i} \frac{\alpha_k^j}{\alpha_k^{\mathrm{merged}}}[\boldsymbol{P}_k^j + (\boldsymbol{u}_k^j - \boldsymbol{u}_k^{\mathrm{merged}})(\boldsymbol{u}_k^j - \boldsymbol{u}_k^{\mathrm{merged}})^{\mathrm{T}}] \tag{18}$$

其中，$\alpha_k^{\mathrm{merged}}$，$\boldsymbol{u}_k^{\mathrm{merged}}$ 和 $\boldsymbol{P}_k^{\mathrm{merged}}$ 分别为合并后获得的混合项的权值、均值和方差。

该融合方法可以保持原混合高斯项的中心矩不变[20]，而且计算方便，被大多数基于合并思想的改进方法采用[8,10-11]。

3 算法分析

3.1 算法流程

为了清晰算法步骤，下文中用 GMM 分别表示初始状态、系统噪声及观测噪声，并采用卡尔曼滤波方法对目标状态进行估计，具体算法流程可参考算法 1。

算法 1　聚类 – 合并算法

Alg. 1　The cluster-merge algorithm

输入：初始状态高斯项个数及参数、系统噪声高斯项个数及参数、测量噪声高斯项个数及参数，各时刻的测量值；

输出：下一时刻高斯混合项个数及相应参数；

开始迭代；
for $k=1,\cdots,t$
　1. 时间更新：$I=0$
　for $i=1,\cdots$，初始状态个数
　　for $l=1,\cdots$，系统噪声个数
　　$I=I+1$；
　　更新预测状态、预测协方差、权值；
　　end for
　end for
　保留当前混合项个数 $I_{k|k-1}=I$；
　2. 观测更新：$II=0$
　for $i=1,\cdots$，预测状态个数
　　for $j=1,\cdots$，测量噪声个数
　　$II=II+1$
　　更新状态估计量、协方差矩阵、权重向量；
　　end for
　end for
　保存混合项 $I_k=II$；
　3. 计算状态及协方差；
　4. 判断是否满足聚类条件，即
　(1)均值的 1.5 倍；若大于则作为下一时刻迭代项返回；
　(2)是否连续两时刻权值都小于阈值 0.01；满足，则直接删掉，并对剩余权值重新归一化；
　5. 利用带约束的非线性优化理论搜索最优置信范围 $\boldsymbol{P}^*$；
　6. 计算判别矩阵并对混合项聚类；
　7. 重新分配交叉项；
　8. 采用多元素合并方法获得各高斯簇的混合项；
　9. 返回剩余混合项并将其作为下一时刻初值进行迭代；
end for

3.2 算法复杂度

在时间复杂度上，所提算法首先需要判断混合项之间是否具有相似分布特性，即计算判别矩阵。而判别矩阵只需得知上三角的值即可判别所有混合项间的相似关系，故第一步的时间复杂度为 $O(n^2)$；第二步因交叉项自带标签，故查找交叉项并不会占用时间，但在重新分配时需判断其与

各所属簇间的最小距离，因此时间复杂度为 $O(n^2)$；第三步因采用多元素融合方法，其时间复杂度为 $O(\lg n)$；第四步为对置信范围采用非线性优化方法寻优，其时间复杂度为 $O(m \times p \times n^2)$，其中 m 为算法迭代次数，p 为搜索粒子数量。由此可见，用于搜索的时间复杂度受代价函数性能、粒子初始状态及数量的影响，并且随着状态维度的升高，用于搜索最优 $\boldsymbol{P}^*$ 值的迭代次数会呈指数级增加。因此所提算法更适用于低维度的状态。而为了限制该步骤的时间复杂度，有效合理地缩窄搜索范围至关重要。

针对空间复杂度，因算法需要空间存储融合后的混合项、高斯簇，故其空间复杂度为 $O(n)$。标准算法因在迭代过程中需要大量空间来存储中间变量，并且每次采样混合项数量呈指数式增长，所以在空间复杂度上，改进算法具有明显优势。

4　性能分析

为了验证算法的有效性，首先利用三种典型场景来确定置信范围的有效搜索域，从而提高算法的搜索效率。同时为了方便图形呈现，选取一维情况加以说明。最后，针对二种常见的目标运动场景对运动目标的状态进行估计，以均方根误差（Root Mean Square Error，RMSE）作为评价跟踪性能的依据，并以中央处理单元（Central Processing Unit，CPU）运行时间作为评价时间复杂度的标准。

4.1　置信范围有效区间

4.1.1　四混合项呈单峰分布情况

四个混合项的参数分别设置如下：

$$\Omega^1(1) = \{0.3, \mathrm{N}(x;1,1^2)\}$$
$$\Omega^1(2) = \{0.1, \mathrm{N}(x;1.5,1.2^2)\}$$
$$\Omega^1(3) = \{0.4, \mathrm{N}(x;2,0.5^2)\}$$
$$\Omega^1(4) = \{0.2, \mathrm{N}(x;3,0.8^2)\}$$

令 P^* 分别为 $1\sigma, 2\sigma, 3\sigma, 4\sigma$ 时，PDF 近似效果如图 1 所示。由图可知，P^* 在 $1\sigma \sim 3\sigma$ 时算法对单峰 PDF 近似效果较好，而在 $P^* = 4\sigma$ 时效果明显变差。这主要是因为当 P^* 选择过大时，无法等效成单峰分布的混合项被归并成一簇，从而导致混合项多样性退化，导致近似精度下降。

表 1 为不同 P^* 下算法效率。由表可知，当 P^* 取 2σ 和 3σ 时近似效果几乎相同，这说明单纯的增加或减小 P^* 并不一定能够改善近似效果，这与当前混合项的分布特性有关。并且，随着 P^* 的增大，最终的混合项个数将呈下降趋势。

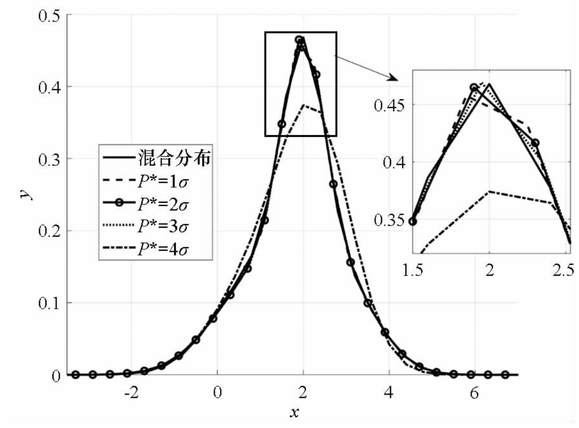

图 1　近似单峰 PDF

Fig. 1　Approximated PDF to single peak

表 1　单峰 PDF 下算法效率

Tab. 1　Efficiency to single-peak PDF

置信范围	高斯混合项个数	计算耗时/s
1σ	4	0.018
2σ	3	0.021
3σ	3	0.021
4σ	2	0.023

4.1.2　四混合项呈双峰分布情况

四个高斯混合项的参数设置如下所示：

$$\Omega^2(1) = \{0.3, N(x;0.5,1^2)\}$$
$$\Omega^2(2) = \{0.1, N(x;2,1.2^2)\}$$
$$\Omega^2(3) = \{0.4, N(x;3.5,0.5^2)\}$$
$$\Omega^2(4) = \{0.2, N(x;4,0.8^2)\}$$

图 2 为置信范围分别为 1σ、2σ、3σ 及 4σ 时对应的 PDF 曲线。由图可以看出 PDF 的近似效果随着置信范围的增加而逐渐变差。当 $P^* = 3\sigma$ 或 $P^* = 4\sigma$ 时，近似效果不佳，尤其在峰值处。

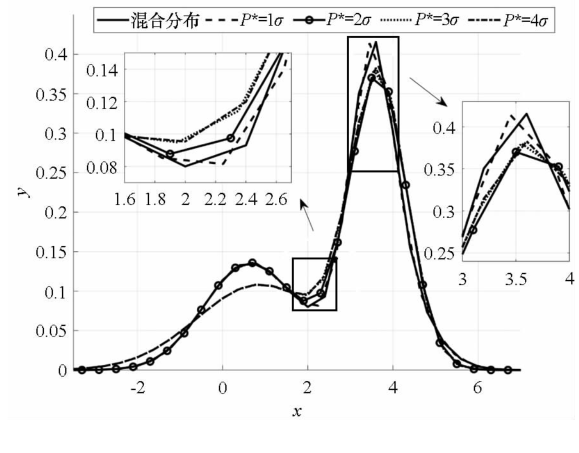

图 2　近似双峰 PDF

Fig. 2　Approximated PDF to double peak

该情况下算法效率如表 2 所示。由表可知，在近似具有双峰分布的 PDF 时，算法在降低高斯项个数方面性能较好，当 $P^*=4\sigma$ 时，其返回的高斯项个数仅为初始输入的一半，但耗时最长。

表 2　双峰下 PDF 算法效率

Tab. 2　Efficiency to double-peak PDF

置信范围	高斯混合项个数	计算耗时/s
1σ	4	0.017
2σ	3	0.026
3σ	2	0.027
4σ	2	0.030

4.1.3　八混合项呈多峰分布情况

八个混合项的参数设置如下：

$$\Omega^3(1)=\{0.1,\mathrm{N}(x;0.5,1)\}$$
$$\Omega^3(2)=\{0.05,\mathrm{N}(x;2,1.2^2)\}$$
$$\Omega^3(3)=\{0.35,\mathrm{N}(x;3.5,0.5^2)\}$$
$$\Omega^3(4)=\{0.1,\mathrm{N}(x;4,0.8^2)\}$$
$$\Omega^3(5)=\{0.1,\mathrm{N}(x;5,0.2^2)\}$$
$$\Omega^3(6)=\{0.2,\mathrm{N}(x;-1,0.3^2)\}$$
$$\Omega^3(7)=\{0.04,\mathrm{N}(x;0,4^2)\}$$
$$\Omega^3(8)=\{0.06,\mathrm{N}(x;1,3^2)\}$$

图 3 为不同置信范围 P^* 下对应的近似 PDF。由图可知在近似多峰分布的 PDF 时，改进算法的整体近似效果较为理想，但随着 P^* 的增大，对于起伏处的 PDF 近似效果逐渐变差。算法效率如表 3 所示。

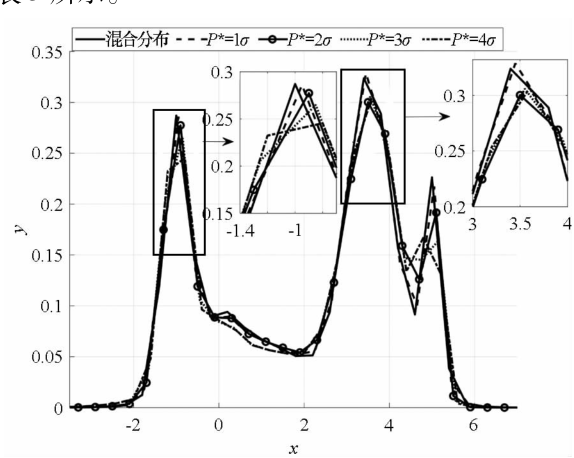

图 3　近似多峰 PDF

Fig. 3　Approximated PDF with multiple peaks

由表 3 可以看出，P^* 的增加使得耗时也稍有增加，但同时获得的高斯混合项的个数也在减少，这更有利于用在递推算法中。

通过分析上述三种典型情况，可知过大的 P^* 会使得近似精度变差，尽管可以大幅度降低输出的混合项个数、节省 GMM 迭代时间，但相应会增加聚类－合并的处理时间。

表 3　多峰 PDF 下算法效率

Tab. 3　Efficiency to multi-peak PDF

置信范围	高斯混合项个数	计算耗时/s
1σ	7	0.041
2σ	6	0.046
3σ	5	0.048
4σ	4	0.051

在通过 EISE 代价函数计算的最优置信范围下，三种典型情况的近似性能如表 4 所示。

表 4　最优置信范围下近似效果

Tab. 4　Efficiency for the best confidence interval

场景	最优置信范围	混合项个数	计算耗时/s
单峰	1.8σ	3	0.028
双峰	1.7σ	2	0.03
多峰	1.6σ	6	0.052

通过上面三个典型的场景，将置信范围 P^* 的搜索域设定在 $[0.5\sigma,3\sigma]$ 以提高算法效率。

4.2　跟踪性能

在雷达接收系统中，因目标的散射特性使观测噪声也可能呈现非高斯性质，称之为闪烁噪声[21]。不同于高斯噪声，闪烁噪声尾部较长，而在中心区域则类似高斯形状。其概率分布可表示为

$$f_t=(1-\varepsilon)f_g+\varepsilon f_l \tag{19}$$

式中，f_t、f_g 及 f_l 分别代表闪烁噪声、高斯噪声及拉普拉斯噪声的分布，其表达式分别为

$$f_g=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x_s^2}{2\sigma^2}\right) \tag{20}$$

$$f_l=\frac{1}{2\eta}\exp\left(-\frac{|x_s|}{\eta}\right) \tag{21}$$

其中：x_s 表示特征变量；ε 为一个小于 1 的小正数，其表示为闪烁频率，下文设 $\varepsilon=0.01$；f_g 的方差 σ 小于 f_l 的方差 η。

设目标状态向量为 $\boldsymbol{x}=[x,\dot{x},\ddot{x}]^{\mathrm{T}}$，其中 x、$\dot{x}$、$\ddot{x}$ 分别为目标位置、速度和加速度信息；初始状态和观测噪声分别用 GMM 表示为

$$\begin{cases}p(\boldsymbol{x}_0)=0.6\times\mathrm{N}(\boldsymbol{x}_0;\boldsymbol{x}_1,\boldsymbol{P}_1)+0.4\times\mathrm{N}(\boldsymbol{x}_0;\boldsymbol{x}_2,\boldsymbol{P}_2)\\ p(\boldsymbol{v}_0)=0.5\times\mathrm{N}(\boldsymbol{v}_0;\boldsymbol{v}_1,\boldsymbol{R}_1)+0.5\times\mathrm{N}(\boldsymbol{v}_0;\boldsymbol{v}_2,\boldsymbol{R}_2)\end{cases}$$

其中参量的设定为：系统状态噪声的方差 $\boldsymbol{Q}=\mathrm{diag}[10,4,2]$，$\boldsymbol{x}_1=[30,20,10]^{\mathrm{T}}$，$\boldsymbol{P}_1=\mathrm{diag}[300,$

$100,50]$, $\boldsymbol{x}_2=[20,10,5]^{\mathrm{T}}$, $\boldsymbol{P}_2=\mathrm{diag}[350,200,40]$, $\boldsymbol{v}_1=[10,5,2]^{\mathrm{T}}$, $\boldsymbol{R}_1=\mathrm{diag}[100,25,4]$, $\boldsymbol{v}_2=[-10,3,1]^{\mathrm{T}}$, $\boldsymbol{R}_2=\mathrm{diag}[100,9,1]$。

该仿真场景下采样周期 $t_s=1$ s，仿真时长 $T_o=200$ s，跟踪模型为交互式多模型（Interacting Multiple Model, IMM），该模型主要由当前“统计”（Current Statistic, CS）模型及已知转弯速率的转弯（Constant Turn, CT）模型组成。其中机动频率 $\alpha=0.01$ Hz，最大加速度 $a_{\max}=50\ \mathrm{m/s^2}$，转弯速率为 0.02π rad/s，CS 模型中的其他参数可参考文献[22]，IMM 模型的转换矩阵为 $\boldsymbol{p}_{\mathrm{IMM}}=\begin{bmatrix}0.95 & 0.05\\ 0.05 & 0.95\end{bmatrix}$；Monte Carlo 仿真次数为 100 次，正则化常数设定为 0.3。对比算法选 GMRC、Runnall 和 GMRP，本文在最优置信范围下的聚类－合并算法简称 GMRCM。

4.2.1　圆周运动场景

在弱机动场景下，目标以 0.02π rad/s 的角速度作匀速圆周运动，其在 100 s 以内位置 RMSE 的对比如图 4 所示。

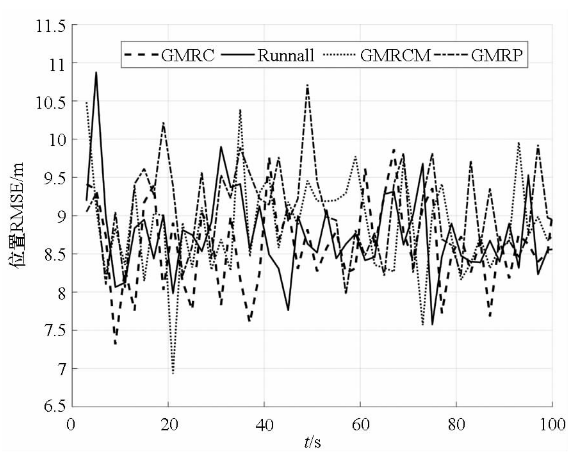

图 4　位置 RMSE

Fig. 4　Position RMSE

图 4 中 GMRC、Runnall、GMRCM、GMRP 四种算法的平均位置 RMSE 分别为 8.96 m，9.29 m，9.32 m 以及 9.78 m，由此可以看出对于弱机动目标，GMRC 算法的跟踪精度最好，其次 Runnall 与本文提出的 GMRCM 算法精度相当，而 GMRP 算法的精度最差。在第 50 s 时刻目标状态的后验分布及混合高斯近似的后验分布如图 5 所示。

由图 5 可知，因多模型及非高斯噪声的存在，估计的位置后验概率密度函数呈现非高斯性质。各算法对于圆周运动的位置估计精度较为理想，其偏差幅度并不大。其中 GMRC、Runnall 以及 GMRCM 算法的估计精度相当，略优于 GMRP

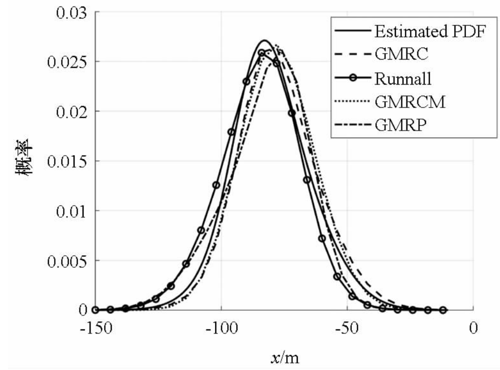

图 5　目标状态后验分布

Fig. 5　Posterior distribution of target state

算法。

置信范围 P^* 随时间的变化情况如图 6 所示。

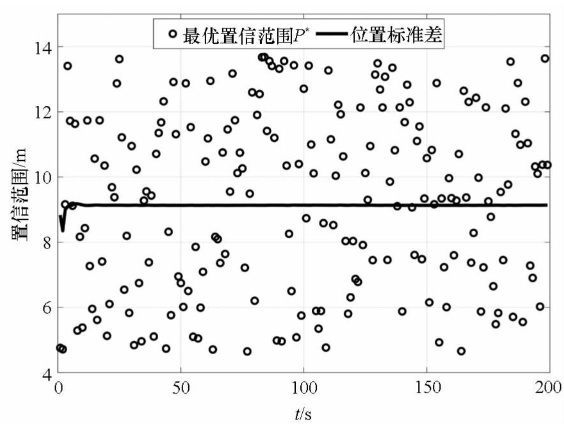

图 6　最优置信范围的变化

Fig. 6　Variation of the confidence interval

由图 6 可知，针对弱机动情况下，P^* 大部分情况处于 $0.5\sim1.5\boldsymbol{P}(1,1)$ 之间（对应图中 4 ~ 14 m），即说明了在目标状态不发生突变的情况下，一般会取较大置信范围值，这主要是由于当前目标的状态后验分布比较均匀，具有多峰分布的情况并不多才使得 P^* 较大。而 P^* 会发生不断的跳动主要是由初始搜索点选择的随机性以及存在多个最优解造成的。

图 7 为间隔 10 个采样时刻的高斯混合项个数的变化情况。

由图 7 知，改进算法所得高斯混合项个数要明显少于另外三种算法，这主要是由于 GMRP 算法与 Runnall 算法都是达到人为设定数量后才停止合并，而 GMRC 算法则还需要进行聚类操作，其数量会少于 Runnall 算法，但其用时则会受到影响。

针对算法的时间复杂性，分别记录了 GMRC、GMRP、GMRCM、Runnall 这四种算法的运行时长，分别为 0.52 s，0.44 s，0.38 s，0.48 s。由此可看出本文算法由于高效的搜索方法，迭代过程中

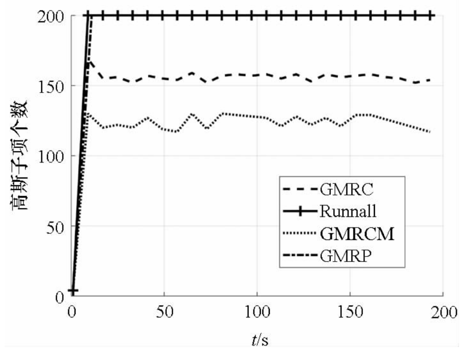

图 7　高斯混合项个数

Fig. 7　Number of Gaussian components

的高斯混合项数量少，耗时最短。而 GMRP 算法由于在删除混合项过程中用时极短，即便在迭代过程中会耗时较长，但仍取得了较优的时间复杂度；GMRC 算法由于计算最为复杂，故其用在筛选合并项过程中耗时最长，时间复杂度较高；Runnall 算法因寻找合并项相对简单，故时间复杂度稍好于 GMRC 算法。

由上述分析可知，本文所提算法虽然在近似精度上不如 GMRC 算法，但能够达到与 Runnall 算法相当的精度，并且要明显强于 GMRP 算法。同时，在时间复杂度上，所提算法在对比算法中效率是最高的。

4.2.2　阶跃机动场景

该场景下目标机动较强，加速度变化情况为

$$\ddot{x}(t)=\begin{cases}10\ \mathrm{m/s^2} & 0<t<50\ \mathrm{s}\\ 20\ \mathrm{m/s^2} & 50\ \mathrm{s}\leqslant t<100\ \mathrm{s}\\ -10\ \mathrm{m/s^2} & 100\ \mathrm{s}\leqslant t\leqslant 200\ \mathrm{s}\end{cases}$$

位置 RMSE 的对比情况如图 8 所示。针对阶跃机动情况，各算法平均位置 RMSE 为：GMRC 为 9.32 m、Runnall 为 9.58 m、GMRCM 为 10.2 m、GMRP 为 12.17 m。由此可知，虽然所提 GMRCM 算法的跟踪精度要明显好于 GMRP 算法，但相较

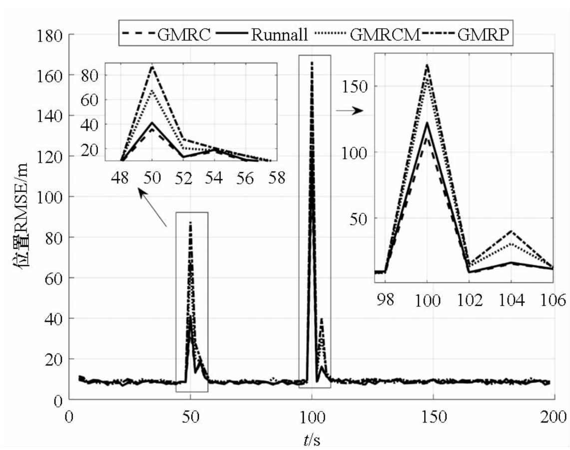

图 8　位置 RMSE

Fig. 8　Position RMSE

于另两个算法，尤其在目标发生阶跃运动时间段内，跟踪精度仍有待提高。

第 50 s 处目标状态后验分布如图 9 所示。由图可知，因目标发生突变机动导致两个模型的滤波状态发生差异，使得模型的状态后验分布呈现双峰分布。从各算法近似曲线不难看出 GMM 可较好地近似出非高斯分布特性，但因状态的突变导致高斯混合子滤波器的滤波精度下降，PDF 逼近程度并不高。但从位置 RMSE 曲线可以看出，精度仍可满足要求，并且 GMRC 与 Runnall 的近似精度较高，GMRCM 次之，而 GMRP 最差。

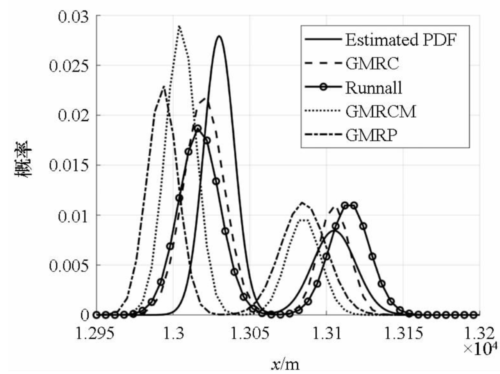

图 9　目标状态后验分布

Fig. 9　Posterior distribution of target state

P^* 变化情况如图 10 所示。由图可以看出，当目标发生阶跃机动时，GMRCM 算法的最优置信范围会变小以适应状态的突变，进而获得较优的跟踪性能。而对于目标运动平缓阶段，最优置信范围则会增大来适应稳定的状态后验分布。

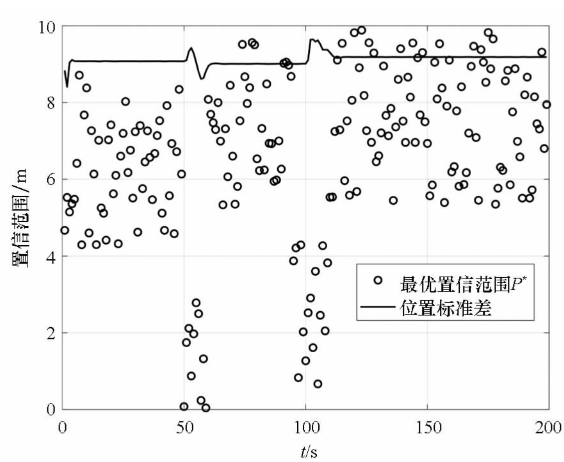

图 10　最优置信范围的变化

Fig. 10　Change of the confidence interval

采样点处的混合项个数如图 11 所示。图中本文所提算法 GMRCM 的高斯混合项个数要少于其他三种算法。尽管当目标发生阶跃机动时，目标的高斯项个数还是会增加，这主要是由于机动时刻目标状态的后验分布呈多峰形式，使得高斯

簇增加进而增多了数量,但同比于其他三种算法,仍有优势。

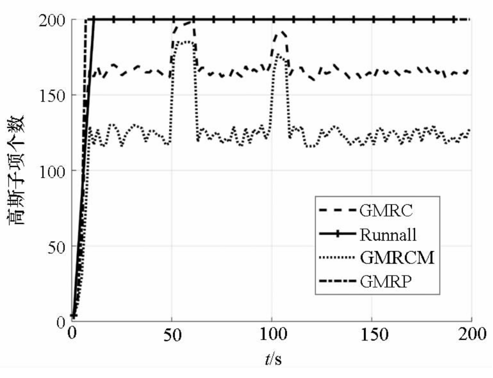

图11 高斯混合项个数

Fig. 11 Number of Gaussian components

针对算法的时间复杂性,GMRC、Runnall、GMRCM、GMRP 四种算法的计算耗时分别为 0.67 s, 0.57 s, 0.45 s, 0.62 s。由此可见针对阶跃机动情况本文算法可以获得较高的跟踪效率。

由以上分析可知,在跟踪阶跃机动目标时,本文所提算法在跟踪精度上较之 GMRC、Runnall 算法稍有不足,但仍在可接受范围内。并且,本文算法的效率在四种对比算法中是最优的。

5 结论

针对 GMM 估计非高斯系统状态时会导致高斯混合项个数呈指数级增长问题,本文基于相似分布特性准则,提出了一种混合聚类－合并方法。通过定义相似分布特性准则,基于 EISE 代价函数寻求最优置信区间以对混合项进行高斯聚类,建立具有不同分布特性的高斯簇。为了解决各高斯簇中的交叉项的复用问题,引入 LNN 思想对交叉项重新分配。为了削减迭代过程中混合项数量,采用并行多元素融合法将高斯簇内的混合项合并。最后通过仿真验证了算法在保持跟踪精度及提高运算效率方面的有效性。

参考文献(References)

[1] Li T C, Prieto J, Corchado J M. A short revisit of nonlinear Gaussian filters: state-of-the-art and some concerns[C]//Proceedings of International Conference on Ubiquitous Wireless Broadband, 2016: 1－4.

[2] Sum W, Yang Y J. Adaptive maneuvering frequency method of current statistical model[J]. IEEE/CAA Journal of Automatica Sinica, 2017, 4(1): 154－160.

[3] Li L Q, Li C L, Cao W M, et al. Fuzzy quadrature particle filter for maneuvering target tracking[J]. International Journal of Fuzzy Systems, 2016, 18(4): 647－658.

[4] Sorenson H W, Alspach D L. Recursive Bayesian estimation using Gaussian sums[J]. Automatica, 1971, 7(4): 465－479.

[5] Pishdad L, Labeau F. Approximate MMSE estimator for linear dynamic systems with Gaussian mixture noise[J]. IEEE Transactions on Automatic Control, 2017, 62(5): 2457－2463.

[6] Pishdad L, Labeau F. A new reduction scheme for Gaussian sum filters[C]//Proceedings of 48th Asilomar Conference on Signals, Systems and Computers, 2015: 1351－1357.

[7] Ali-Löytty S. Gaussian mixture filters in hybrid positioning[D]. Finland: Tampere University of Technology, 2009.

[8] Williams J L, Maybeck P S. Cost-function-based hypothesis control techniques for multiple hypothesis tracking[J]. Mathematical & Computer Modelling, 2004, 43(9): 976－989.

[9] Ardeshiri T, Orguner U, Özkan E. Gaussian mixture reduction using reverse Kullback-Leibler divergence[J]. arXiv: 1508.05514, 2015.

[10] Runnalls A R. Kullback-Leibler approach to Gaussian mixture reduction[J]. IEEE Transactions on Aerospace Electron System, 2007, 43(3): 989－999.

[11] Schieferdecker D, Huber M F. Gaussian mixture reduction via clustering[C]//Proceedings of 12th International Conference on Information Fusion, 2009: 1536－1543.

[12] Crouse D F, Willett P, Pattipati K, et al. A look at Gaussian mixture reduction algorithms[C]//Proceedings of 14th International Conference on Information Fusion, 2011: 1－8.

[13] Cai L F, Tian X M, Chen S. Monitoring nonlinear and non-Gaussian processes using Gaussian mixture model-based weighted kernel independent component analysis[J]. IEEE Transactions on Neural Networks & Learning Systems, 2017, 28(1): 122－135.

[14] Luo Z H, He W S, Li W M, et al. Real-time detection algorithm of abnormal behavior in crowds based on Gaussian mixture model[C]//Proceedings of 12th International Conference on Computer Science and Education, 2017: 183－187.

[15] Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction[C]//Proceedings of 17th International Conference on Pattern Recognition, 2004: 28－31.

[16] Reynolds D A, Quatieri T F, Dunn R B. Speaker verification using adapted Gaussian mixture models[J]. Digital Signal Processing, 2000, 10(1/2/3): 19－41.

[17] Cui M J, Feng C, Wang Z K, et al. Statistical representation of wind power ramps using a generalized Gaussian mixture model[J]. IEEE Transactions on Sustainable Energy, 2018, 9(1): 261－272.

[18] Hennig C. Methods for merging Gaussian mixture components[J]. Advances in Data Analysis & Classification, 2010, 4(1): 3－34.

[19] Psiaki M L. The "Blob" filter: Gaussian mixture nonlinear filtering with re-sampling for mixand narrowing[J]. IEEE Transactions on Signal Processing, 2016, 64(21): 5499－5512.

[20] Terejanu G A. An adaptive split-merge scheme for uncertainty propagation using Gaussian mixture models[C]//Proceedings of 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2011.

[21] 宋小全, 孙仲康. 非高斯噪声下的机动目标跟踪[J]. 电子学报, 1998, 26(9): 40－46.
SONG Xiaoquan, SUN Zhongkang. Maneuvering target tracking with non-Gaussian noise[J]. Acta Electronica Sinica, 1998, 26(9): 40－46. (in Chinese)

[22] 黄伟平, 徐毓, 王杰. 基于改进"当前"统计模型的转弯机动跟踪算法[J]. 控制与决策, 2011, 26(9): 1412－1416.
HUANG Weiping, XU Yu, WANG Jie. Algorithm based on modified current statistic mode for turn maneuver[J]. Control and Decision, 2011, 26(9): 1412－1416. (in Chinese)