

1-8-2019

Message Passing Algorithm through Particles on Factor Graph in Cooperative Positioning Network

Xinyue Fan

Chongqing Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

Wang Guan

Chongqing Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

Zhou Fei

Chongqing Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Message Passing Algorithm through Particles on Factor Graph in Cooperative Positioning Network

Abstract

Abstract: Combining the sum-product algorithm with the factor graph can achieve cooperative positioning by a distributed manner. The sum-product algorithm is a message passing algorithm. However, the parameter method cannot meet the positioning needs because of the large error in the non-linear non-Gaussian environment. Therefore, this paper presents an algorithm to achieve message passing in the form of particle. The calculation of the message in the factor consists of summation and quadrature processes. *The proposed method obtains the sum of messages by importance sampling, obtains the product of messages using Gibbs sampling. Particle-based information passing algorithm can simplify the complex joint posterior probability distribution. Compared with message passing algorithm based on parameters, the method based on particles improves the positioning accuracy in non-linear, non-Gaussian environments.*

Keywords

distributed algorithms, factor graph, sum-product algorithm, message passing algorithm, particle

Recommended Citation

Fan Xinyue, Wang Guan, Zhou Fei. Message Passing Algorithm through Particles on Factor Graph in Cooperative Positioning Network[J]. Journal of System Simulation, 2018, 30(8): 2900-2907.

协作网络因子图中粒子形式的信息传递算法

范馨月, 王冠, 周非

(光通信与网络重点实验室, 重庆邮电大学, 重庆 400065)

摘要: 和积算法结合因子图可以用分布式方式实现协作定位。和积算法是一种信息传递算法, 然而, 在非线性和非高斯环境下用参数法实现信息传递误差较大, 不能满足定位需要, 提出一种算法用粒子形式来实现信息传递。因子图中的信息计算包括求和与求积两个过程。粒子形式的信息传递算法利用重要性采样得到求和信息, 利用吉布斯采样得到求积信息。提出的算法能简化复杂的网络节点的联合后验概率。与基于参数的信息传递算法相比, 粒子形式表示方法提高了在非线性和非高斯环境下的定位精度。

关键词: 分布式算法; 因子图; 和积算法; 信息传递算法; 粒子

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2018) 08-2900-09

DOI: 10.16182/j.issn1004731x.joss.201808011

Message Passing Algorithm through Particles on Factor Graph in Cooperative Positioning Network

Fan Xinyue, Wang Guan, Zhou Fei

(Chongqing Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications,
Chongqing 400065, China)

Abstract: Combining the sum-product algorithm with the factor graph can achieve cooperative positioning by a distributed manner. The sum-product algorithm is a message passing algorithm. However, the parameter method cannot meet the positioning needs because of the large error in the non-linear non-Gaussian environment. Therefore, this paper presents an algorithm to achieve message passing in the form of particle. The calculation of the message in the factor consists of summation and quadrature processes. The proposed method obtains the sum of messages by importance sampling, obtains the product of messages using Gibbs sampling. Particle-based information passing algorithm can simplify the complex joint posterior probability distribution. Compared with message passing algorithm based on parameters, the method based on particles improves the positioning accuracy in non-linear, non-Gaussian environments.

Keywords: distributed algorithms; factor graph; sum-product algorithm; message passing algorithm; particle

引言

有效的位置信息对当前和未来的移动终端应



收稿日期: 2016-09-23 修回日期: 2017-02-02;
基金项目: 国家自然科学基金(61471077);
作者简介: 范馨月(1979-), 女, 四川犍为, 硕士, 副教授, 研究方向为认知无线电、通信信号处理、图像处理等; 王冠(1988-), 男, 山西临汾, 硕士生, 研究方向为无线定位。

用是必要的。无线传感网络中节点的位置收集、节点之间的位置信息交换对节点应用是至关重要的。传统的定位方法对锚节点具有很高的依赖性, 每个网络节点只能通过相对于锚节点的测量值来估计自己的位置并且受锚节点数量的限制。协作定位是一种较新的定位方法^[1], 该方法在移动节点之间引入协作有助于节点估计自己的位置。当移动节点仅

<http://www.china-simulation.com>

• 2900 •

仅根据锚节点无法估计出自己位置时,协作能够使移动节点完成定位。

定位和导航算法的目标就是根据测量值和一些先验信息来估计自己的位置。文献[2-5]提出一些关于参数的估计和统计推理方法,其中一些方法可以被应用到定位中来估计位置参数。众所周知,在高斯噪声的线性系统下卡尔曼滤波的性能较高^[6]。但是,在实际环境中卡尔曼滤波的条件不能得到满足。贝叶斯方法求解状态估计问题的方法是通过用已获得的观测来建立当前状态的后验概率密度函数。我们将关于节点位置的后验概率密度函数 $p(x|z)$ 称为位置置信度^[7-8](Positional Belief, PB)。只要我们获得任何一个节点的 PB 分布,就可以通过求均值或求众数来分别得到一个关于节点位置的最小均方误差估计和最大后验概率密度估计。实现协作的方式有 2 种:集中式结构和分布式结构^[9]。集中式结构拥有一个集中处理中心根据所有节点的测量值和先验信息来更新所有节点的联合 PB。但是由于信息传输的通信负担和高维度的联合 PB 带来的较高计算量使得集中式结构不适用于协作定位,通常我们采用分布式结构来实现协作。实现分布式协作定位的一种常用方法是和积算法(Sum-product Algorithm, SPA)^[10-11],它是基于因子图(Factor Graph, FG)的。SPA 是一种信息传递算法(Message Passing Algorithm, MPA),被广泛应用到无线传感网络中的参数估计问题。粒子滤波是解决非线性、非高斯问题的一种有效方法,它是一种基于蒙特卡洛的方法,用随机采样得到的样本和与每个样本点所对应的权值来表示节点 PB。然而,传统粒子滤波在建立采样样本、更新采样样本时都是仅仅基于系统模型和一些先验信息的,而没有在邻节点之间进行协作。文献[12]提出的信息传递算法是基于参数的,称为(Parameter-based MPA, PA-MPA)。PA-MPA 将 FG 中任何节点信息都用一个高斯分布的均值与方差来表示,被简化为参数传递。PA-MPA 的计算复杂度低,不需要对节点信息做任何处理,但是在非线性、非高斯环境中

PA-MPA 的定位误差较大。本文提出一种粒子形式的信息传递算法(Particle-based MPA, P-MPA)。

P-MPA 利用粒子的形式来表示每个节点在每个时刻的信息,并且运用 SPA 进行信息更新。因子图中各个顶点信息的计算包括两个过程:求和与求积过程也可称作预测和修正过程。文献[13-15]中的信息传递算法在预测阶段根据上一时刻信息的粒子形式计算当前时刻的信息,然后构造重要性函数对当前时刻的信息进行采样,得到当前时刻信息的粒子形式。P-MPA 在预测阶段直接运用系统模型进行重要性采样得到预测信息粒子形式,在修正阶段对来自所有测量值的信息积进行吉布斯采样得到该时刻节点估计信息的粒子形式。吉布斯采样能够避免对复杂的信息积进行采样的操作。与粒子滤波相比 P-MPA 算法在修正阶段不仅能够利用来自邻居未知节点(邻节点中的未知节点)的信息还能够利用邻居锚节点(邻节点中的锚节点)的信息。在锚节点分布不足或者用户节点接受范围较小的场景中,例如矿井和偏僻的野外,传统定位算法包括全球定位系统(Global Positioning System, GPS)和信标定位将会由于接受不到足够的测量信号而不能完成定位。信标定位指用陆地通信基站实现定位的方法。P-MPA 算法用粒子形式实现了协作定位,可以在用户之间实现信息共享,提高了定位成功率与定位精度。

1 因子图与信息传递算法

由于本文算法是基于因子图 FG 来实现的,因此我们首先介绍因子图的作用,然后介绍如何根据信息表达式得到因子图,最后介绍利用和积算法计算因子图中各顶点的信息。

1.1 因子图

因子图能够用图解模型来代表变量之间的关系,将一个复杂的网络结构模型化。更重要的是由于网络节点的联合 PB 比较复杂,因子图结合 SPA 算法可以将网络节点的联合 PB 进行边缘化,从而

将复杂的实际问题简单化。假设某个网络的联合概率密度函数 PB 可以被因式分解, 结合网络 FG 中节点之间的独立关系, 能够实现每个网络节点独自更新自身的 PB。设某网络中节点的联合 PB 可以被分解为 M 项, 则式(1)可表示为:

$$p_{x|z}(x|z) = \prod_{k=1}^M f_k(x_k) = f_A(x_1)f_B(x_1, x_2)f_C(x_1, x_4)f_D(x_2, x_3)f_E(x_4, x_5) \quad (1)$$

我们需要用 FG 的形式来表示因式分解, 根据因式分解结果来得到 FG 的步骤如下:

- (1) 每一个因子对应一个顶点, $f_k(\cdot)$ 代表因子信息, 在因子图中我们用正方形表示;
- (2) 每一个变量对应一个顶点, x_k 代表变量信息, 在因子图中我们用圆形表示;
- (3) 每条边用来连接有对应关系的变量顶点和因子顶点。

利用以上三个原则我们得到包含 5 个变量顶点和 5 个因子顶点的因子图, 见图 1。

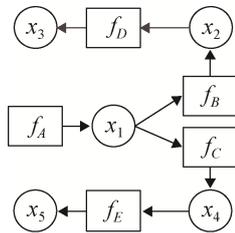


图 1 节点因子图
Fig. 1 Node factor graph

1.2 利用和积算法实现信息传递

SPA 是一种解决在分布式结构中信息传递的有效方法, 用来计算因子图中各个顶点的信息。SPA 计算因子图中各个顶点信息时, 传递的信息将被分为两类: 因子顶点到变量的信息 $\mu_{f \rightarrow x}(\cdot)$ 、变量到因子顶点的信息 $\mu_{x \rightarrow f}(\cdot)$ 。

我们首先计算因子到变量顶点的信息。如图 2 所示, 假设因子图包含一个因子 f 和 $M+1$ 个变量 $\{x_1, x_2, \dots, x_M, x_{M+1}\}$ 。我们将变量 $\{x_1, x_2, \dots, x_M\}$ 当作因子 f 的输入变量, 因子 f 到变量 x_{M+1} 的信息表示为 $\mu_{f \rightarrow x_{M+1}}(x_{M+1})$ 。

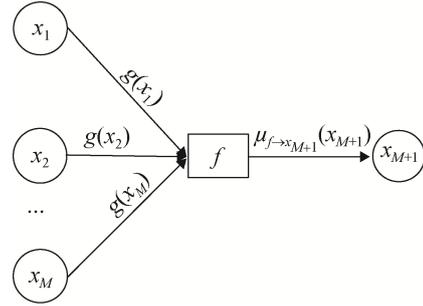


图 2 因子到变量的因子图
Fig. 2 Factor graph of factor to variable

根据因子图 2 建立因子到变量的信息传递方程为:

$$\mu_{f \rightarrow x_{M+1}}(x_{M+1}) = \int f(x_1 \dots x_M, x_{M+1}) g_1(x_1) \dots g_M(x_M) dx_1 \dots dx_M \quad (2)$$

式中: $g(x_i)$ 表示每个变量对因子 f 的输入信息。

假设某一个因子图包含一个变量 x 和 $L+1$ 个因子 $f_1, f_2, \dots, f_L, f_{L+1}$, 所有因子都与变量相连接, 因子与因子之间没有连接, 如图 3 所示。

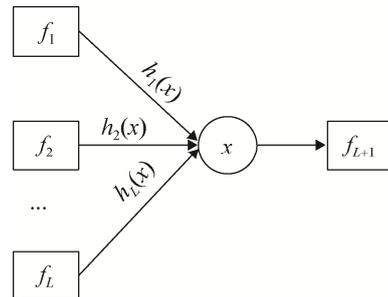


图 3 变量到因子的因子图
Fig. 3 Factor graph of variable to factor

我们将因子 $\{f_1, f_2, \dots, f_L\}$ 当作变量的输入因子, 变量 x 到因子 f_{L+1} 的信息表示为 $\mu_{x \rightarrow f_{L+1}}(x)$ 。根据因子图建立变量到因子的信息传递方程为:

$$\mu_{x \rightarrow f_{L+1}}(x) = \prod_{l=1}^L h_l(x) \quad (3)$$

式中: h_l 表示每个因子对变量的输入信息。

1.3 网络节点的信息分解及传递

在一个协作网络中我们需要建立整个协作网络的 FG, 并且求出每个顶点的信息, 进而得到每个网络节点的 PB。结合协作定位网络的特点, 网

络 FG 中与变量 x_i 相关联的因子分为 2 类: 预测因子和修正因子。其中修正因子又分为 2 类: 与自身测量值相关的信息 $z_{i,self}$ 和与邻节点相关的信息 $z_{i \leftrightarrow j}$ (j 属于 i 节点的一个邻节点)。而与每个因子相关联的变量只有两个: $x_i^{(t)}$ 和 $x_i^{(t+1)}$ 。如图 4 所示, 根据以上信息可以建立协作网络节点的 FG。

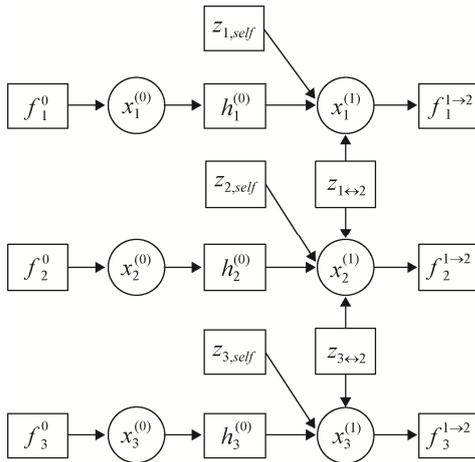


图 4 协作网络节点因子图

Fig. 4 Node factor graph of cooperative network

接下来我们需要求出协作定位网络中每个顶点在每个时刻的信息。假设观测序列和估计参数序列分别为: $\mathbf{Z}^{(1:t)} = [z^{(1)}, z^{(2)}, \dots, z^{(t)}]$ 、 $\mathbf{X}^{(1:t)} = [x^{(1)}, \dots, x^{(t)}]$, 并且观测序列和估计参数序列都具有马尔可夫性。根据贝叶斯定理, 每个节点 PB 更新可被分为 2 步:

(1) 预测

$$p(x^{(t)} | \mathbf{Z}^{(1:t-1)}) = \int p(x^{(t)} | x^{(t-1)}) p(x^{(t-1)} | \mathbf{Z}^{(1:t-1)}) dx^{(t-1)} \quad (4)$$

(2) 修正

$$p(x^{(t)} | \mathbf{Z}^{(1:t)}) = \int p(x^{(t)}, x^{(t-1)} | \mathbf{Z}^{(1:t)}) dx^{(t-1)} \propto p(z^{(t)} | x^{(t)}) p(x^{(t)} | \mathbf{Z}^{(1:t-1)}) \quad (5)$$

根据各个节点状态量的马尔可夫性可以得到一个序列的后验概率密度为:

$$p(\mathbf{X}^{(0:T)} | \mathbf{Z}^{(1:T)}) \propto p(x^{(0)}) \prod_{t=1}^T p(x^{(t)} | x^{(t-1)}) p(z^{(t)} | x^{(t)}) \quad (6)$$

根据网络节点不同类型测量值之间的独立性

和序列的马尔可夫性得到:

$$\prod_{t=1}^T p(z^{(t)} | x^{(t)}) = \prod_{t=1}^T p(z_{self}^{(t)} | x^{(t-1)}, x^{(t)}) p(z_{rel}^{(t)} | x^{(t)}) \quad (7)$$

结合式(6)~(7), 可得:

$$p(\mathbf{X}^{(0:T)} | \mathbf{Z}^{(1:T)}) \propto p(x^{(0)}).$$

$$\prod_{t=1}^T p(x^{(t)} | x^{(t-1)}) p(z_{self}^{(t)} | x^{(t-1)}, x^{(t)}) p(z_{rel}^{(t)} | x^{(t)}) \quad (8)$$

至此网络中任何一个节点的信息表达式可以被因式分解为式(8), 结合网络因子图 4 可以取预测因子 $h_i^{(t)}(x_i^{(t)}, x_i^{(t-1)}) = p(x_i^{(t)} | x_i^{(t-1)})$ 。图 4 网络中有 3 个节点, 我们以节点 2 为研究对象, 建立节点 2 的 2 个信息方程:

(1) 因子到变量的信息方程:

$$\begin{aligned} \mu_{h_2^{(0)} \rightarrow x_2^{(1)}}(x_2^{(1)}) &\propto \\ \int h_2^{(0)}(x_2^{(0)}, x_2^{(1)}) \mu_{f_2^0 \rightarrow x_2^{(0)}}(x_2^{(0)}) dx_2^{(0)} &= \\ \int p(x_2^{(1)} | x_2^{(0)}) p(x_2^{(0)}) dx_2^{(0)} &\quad (9) \end{aligned}$$

式中: $p(x_2^{(0)})$ 为节点 2 的先验信息。

(2) 变量到因子的信息方程:

$$\begin{aligned} \mu_{x_2^{(1)} \rightarrow f_2^{(1 \rightarrow 2)}}(x_2^{(1)}) &= \\ \mu_{h_2^{(0)} \rightarrow x_2^{(1)}}(x_2^{(1)}) \mu_{z_{self}}(x_2^{(1)}) \prod_{j \in S_{\rightarrow 2}^{(1)}} \mu_{j \rightarrow 2}(x_2^{(1)}) &= \\ \mu_{h_2^{(0)} \rightarrow x_2^{(1)}}(x_2^{(1)}) \mu_{z_{self}}(x_2^{(1)}) z_{1 \rightarrow 2}(x_2^{(1)}) z_{3 \rightarrow 2}(x_2^{(1)}) &\quad (10) \end{aligned}$$

式中: $j \in S_{\rightarrow i}^{(t)}$ 表示 i 节点的邻节点集合。式(9)和式(10)分别称为预测和修正。

2 信息传递的实现

由于 SPA 算法只是提供了一种因子图中的信息传递的方法, 并没有提供实现方法。因此, 本节首先设定一个运动的目标, 并给出了目标的运动模型与观测模型; 然后给出了基于参数的信息传递算法, 并分析了它的不足; 最后给出了粒子形式信息传递算法。

2.1 系统模型与问题描述

将一个连续运动的物体作为研究目标, 假设状态量为四维, 用 $\mathbf{X}_k = [x_k \dot{x}_k y_k \dot{y}_k]^T$ 来表示目标在 k 时刻的状态, 其中位置信息为 $p_k = [x_k \ y_k]^T$, 速

度信息为 $v_k = [\dot{x}_k \ \dot{y}_k]^T$ 。假定目标做匀速直线运动，此外目标在运动过程中受到一个随机扰动 $U(k)$ ，称为过程噪声，其方差 Q 越大，目标运动轨迹越曲折。系统状态方程为：

$$\mathbf{X}_k = \mathbf{F}\mathbf{X}_{k-1} + \mathbf{G}U_{2 \times 1}(k) \quad (11)$$

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}$$

观测方程为：

$$\mathbf{z}(k) = \mathbf{H}\mathbf{X}_k + v(k) \quad (12)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

式中： $v(k)$ 为测量噪声。

2.2 基于参数的信息传递

2.2.1 因子到变量顶点的信息传递

假设两个连续的随机变量 x 和 y ， y 的边缘概率密度函数可以通过对 x 和 y 的联合概率密度函数 $f(x, y)$ 对 x 进行边缘积分求得。假设 x 和 y 都服从高斯分布，根据文献[16]可以得到信息方程：

$$\int_{-\infty}^{\infty} \mathcal{N}(y; \alpha x, \sigma_2^2) \mathcal{N}(x; m, \sigma_1^2) dx \propto \mathcal{N}(y; \alpha m, \sigma_1^2 + \sigma_2^2) \quad (13)$$

从式(9)可得因子到变量顶点的信息实际为预测操作，当前时刻状态值与下一时刻的预测值有一个定量关系。假设某个顶点的输入信息服从高斯分布 $\mathcal{N}(x_i^{(0)}; \hat{x}_i^{(0)}, \hat{\sigma}_i^{(0)})$ ，其中 $\hat{x}_i^{(0)}$ 、 $\hat{\sigma}_i^{(0)}$ 分别为节点 i 在 0 时刻估计值的均值和方差。则 0 时刻因子到变量顶点的信息可以表示为：

$$\int_{-\infty}^{\infty} \mathcal{N}(x_i^{(1)}; \alpha x_i^{(0)}, \hat{\sigma}_i^{(1)}) \mathcal{N}(x_i^{(0)}; \hat{x}_i^{(0)}, \hat{\sigma}_i^{(0)}) dx_i^{(0)} \propto \mathcal{N}(x_i^{(1)}; \alpha \hat{x}_i^{(0)}, \hat{\sigma}_i^{(0)} + \hat{\sigma}_i^{(1)}) \quad (14)$$

式中： $\hat{\sigma}_i^{(1)}$ 为 i 节点从 0 时刻到 1 时刻的预测值方差， $\alpha x_i^{(0)}$ 表示节点 i 在 1 时刻的预测值， α 为预测因子，与系统模型式(11)中的 F 矩阵有关。

2.2.2 变量到因子顶点的信息传递

根据式(10)可得变量到因子顶点信息为多个

输入信息乘积的形式。假设每个输入信息服从高斯分布 $\mathcal{N}(x_i; m_i, \sigma_i^2)$ ，则 K 个高斯分布的积为^[17]：

$$\prod_{i=1}^K \mathcal{N}(x_i; m_i, \sigma_i^2) \propto \mathcal{N}(x; \hat{m}, \hat{\sigma}^2) \quad (15)$$

式中：输出信息均值和方差的计算方法根据 K 不同而不同。在 $K=2$ 情况下，均值与方差计算式为：

$$\hat{m} = \frac{\sigma_2^2 \cdot m_1 + \sigma_1^2 \cdot m_2}{\sigma_1^2 + \sigma_2^2}, \hat{\sigma}^2 = \frac{\sigma_1^2 \cdot \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (16)$$

根据以上 2 类信息的计算可得：基于参数的信息传递算法将每个顶点在任何时刻的信息都假设为高斯分布；在计算每个时刻每个节点的信息时只需计算高斯分布的均值和方差，因此基于参数的信息传递算法也称为参数传递算法。

2.3 基于粒子的信息传递

2.3.1 因子到变量顶点的信息传递

我们用蒙特卡洛方法来估计网络 \mathbf{FG} 中每个顶点的信息。在协作定位网络中如图 4 所示，因子到变量顶点的信息对应由 $t-1$ 时刻到 t 时刻的预测。结合图 4，协作定位网络 \mathbf{FG} 在计算因子顶点到变量顶点的信息时，因子顶点的输入信息只有一个。将输入信息 \mathbf{PB} 用 N 个独立同分布的样本和权值来表示 $x^{(t-1)} = \{x_i, \omega_i; 1 \leq i \leq N\}$ ，我们要用输入信息的粒子形式结合因子信息来得到输出信息的粒子形式。根据式(9)可知因子顶点到变量顶点的信息可以表示为一个积分形式：

$$\mu_{f \rightarrow x}(x^{(t)}) = \int \mu_{f \rightarrow x}(x^{(t-1)}) h(x^{(t)}, x^{(t-1)}) dx^{(t-1)} \quad (17)$$

式中： $\mu_{f \rightarrow x}(x^{(t-1)})$ 作为 $x^{(t-1)}$ 的分布函数。我们使用蒙特卡洛理论对式(17)表示的积分形式进行估计。式(17)的积分形式可以看作是 $h(x^{(t-1)})$ 的期望，并且该期望可以用服从分布 $\mu_{f \rightarrow x}(x^{(t-1)})$ 的 N 个独立同分布样本表示。因此，预测信息可表示为：

$$\mu_{f \rightarrow x}(x^{(t)}) = \frac{1}{N} \sum_{i=1}^N h(x^{(t)}, x_i^{(t-1)}) \quad (18)$$

$$x_i^{(t-1)} \sim \mu_{f \rightarrow x}(x^{(t-1)})$$

在得到预测信息 $\mu_{f \rightarrow x}(x^{(t)})$ 后，我们需要建立

该信息的粒子形式。取预测因子 $h(x^{(t)}, x^{(t-1)}) = p(x^{(t)} | x^{(t-1)})$, 因此在对预测信息进行采样的过程相当于对输入信息 $\mu_{f \rightarrow x}(x^{(t-1)})$ 进行重要性采样^[18], 并且重要性密度函数为 $p(x^{(t)} | x^{(t-1)})$ 。具体过程可表示为一个重要性采样过程。采样过程的输入输出形式分别为: $\mu_{f \rightarrow x}(x^{(t-1)})$ 、 $\mu_{f \rightarrow x}(x^{(t)})$; 在得到输入信息后将其表示为粒子形式 $\{x_i^{(t-1)}, \omega_i^{(t-1)}\}_{i=1}^N$ 并进行时间更新, 在时间更新时粒子服从分布: $x_i^{(t)} \sim p(x^{(t)} | x_i^{(t-1)})$; 在时间更新之后需要计算每个粒子的权值 $\omega_i^{(t)} = p(z^{(t)} | x_i^{(t)})$, 并进行归一化处理 $\tilde{\omega}_i^{(t)} = \omega_i^{(t)} / \sum_{i=1}^N \omega_i^{(t)}$, 至此就完成了重要性采样过程。

2.3.2 变量到因子顶点的信息传递

协作定位网络中变量到因子顶点的信息传递相当于目标节点与邻节点之间的协作。其中变量点 x_i 的输入因子包括来自邻节点的测量值 $z_{i,rel}$ 和目标节点该时刻相对于上时刻的位置测量值 $z_{i,self}$ 。假设每个目标节点在修正过程中修正因子数为 L , 并且每个输入因子都可以被表示为粒子形式 $h_l(x) = \{x_l^i, \omega_l^i\}, 1 \leq i \leq N$ 。根据信息转移方程(10) 可得由变量顶点到因子顶点的信息传递方程为:

$$\mu_{x \rightarrow f_{L+1}}(x) = \prod_{l=1}^L \sum_{i=1}^N \omega_l^i x_l^i \quad (19)$$

其中, 式(19)为 L 个高斯混合积, 对该展开式进行采样, 建立它的粒子形式需要 N^L 步操作, 该方法是不切实际的。因此, 我们使用吉布斯采样从该混合积中来获得输出信息的渐进无偏样本。我们将 L 个输入信息表示为高斯分布形式, 均值为 x_l^i , 权值为 ω_l^i , 为了简化问题我们取所有的方差为同一个值 $N^{1/6}$ 。因此式(19)可以重写为:

$$\mu_{x \rightarrow f_{L+1}}(x) = \prod_{l=1}^L \sum_{i=1}^N \omega_l^i \mathcal{N}(x; x_l^i, \sigma_l^i) \quad (20)$$

根据文献[19], L 个高斯混合积仍然为高斯分布:

$$\prod_{i=1}^L \mathcal{N}(x; \mu_i, \sigma_i) \propto \mathcal{N}(x; \bar{\mu}, \bar{\sigma}) \quad (21)$$

$$\bar{\omega} = \frac{\prod_{i=1}^L \mathcal{N}(x; \mu_i, \sigma_i)}{\mathcal{N}(x; \bar{\mu}, \bar{\sigma})} \quad (22)$$

式中: $\bar{\sigma}^{-1} = \sum_{i=1}^L \sigma_i^{-1}, \bar{\sigma}^{-1} \bar{\mu} = \sum_{i=1}^L \sigma_i^{-1} \mu_i$ 。

在吉布斯采样^[20]过程中, 将 L 个高斯分布 $\mathcal{N}(x; \mu_i, \sigma_i) (i=1 \dots L)$ 的粒子形式作为输入信息。 $\{l_j\}_{j=1}^L$ 表示 L 个高斯分布中第 j 个, 并且 $l_j \in \{1, 2, \dots, N\}, j=1, \dots, L$, 表示 j 高斯分布中的某个粒子。在每次迭代过程中我们使 $L-1$ 个高斯分布中的元素 $\{l_k\}_{k \neq j}$ 是固定的, 第 L 个粒子 l_j 根据权值大小来选择。根据式(21)~(22)求出选中的 L 个高斯元素的平均均值 $\bar{\mu}^{(i)}$ 、方差 $\bar{\sigma}^{(i)}$ 和权值 $\bar{\omega}^{(i)} = \omega_j^{(i)} \frac{\mathcal{N}(x; \bar{\mu}, \bar{\sigma}) \mathcal{N}(x; \mu_j^{(i)}, \sigma_j^{(i)})}{\mathcal{N}(x; \bar{\mu}^{(i)}, \bar{\sigma}^{(i)})}$, 根据更新后的权值 $\bar{\omega}^{(i)}$ 来重新选择下一步迭代中 L 个粒子。相同的操作进行 κ 次迭代即可得到采样粒子的分布 $x_i \sim \mathcal{N}(x; \bar{\mu}, \bar{\sigma})$ 。

3 仿真结果

我们将一个连续运动的物体作为研究目标, 目标按照式(11)的方程在二维平面内运动, 观测量按照式(12)的观测模型来获得。假设所有算法在初始时刻的估计值都与真实值相同, 设定目标运动的初始状态为 $X_0 = [-100, 2, 200, 20]^T$ 。根据系统模型可知, 目标做变加速曲线运动, 过程噪声方差 Q 反映的是加速度的波动程度, 过程噪声方差越大运动轨迹越曲折。仿真范围为一个二维平面, 并且仿真环境大小不是固定不变的, 而是根据目标在每次仿真过程中在 x 轴和 y 轴方向上运动的最大位移而定。仿真工具为 MATLAB-7.11.0(R2010b)。P-MPA 算法能够提高定位精度的关键原因是在求变量顶点到因子顶点的信息时不但利用目标节点与上一时刻状态的相对测量值还利用了邻节点的协作信息来修正预测值。因此必须保证在每个时刻目标附近都有一定数量的邻节点供目标节点协作、参考。为了保证每个时刻目标附近都有足够的邻节点, 我们让 100 个邻节点均匀分布在目标节点运动范围

内。考虑到协作定位的实际应用场景中不可能有无限多的邻节点来供目标节点进行协作。因此，本文算法利用节点的通信范围 R_0 来限制邻节点数量。本文中所有信息传递算法都取 R_0 为 100 来进行修正操作。

由于 Kalman 滤波在线性高斯环境中对目标进行连续定位时表现出了较好的性能，而且 Kalman 滤波实现简单，因此本文将同样实现简单的基于参数的信息传递算法与 Kalman 进行对比。首先对比 Kalman 滤波与 PA-MPA 算法的定位性能。在对两种算法进行对比仿真时，取过程噪声方差为 4、观测噪声方差为 100。目标运动过程中设定的过程噪声较小，因此目标的运动轨迹比较平滑。图 5 为用 Kalman 滤波与 PA-MPA 算法的定位结果，反映了目标实际运动轨迹与估计轨迹的偏离程度。

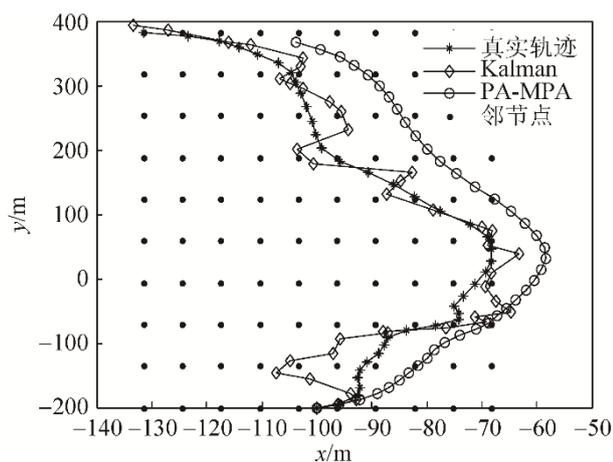


图5 卡尔曼和 PA-MPA 算法的定位结果

Fig. 5 The positioning results of Calman and PA-MPA algorithm

可以看出，Kalman 滤波算法定位误差要远小于基于参数的信息传递算法。Kalman 滤波算法估计点形成的轨迹分布在真实轨迹的两侧，而 PA-MPA 算法估计点形成的轨迹只分布在真实轨迹的一侧。此外，由于观测系统中不包括测量速度、加速度的惯性导航设备，因此观测无法获得速度值。PA-MPA 算法中，目标在与邻节点进行协作阶段只能根据上一时刻的信息和当前邻节点的位置信息修正自身位置信息，而无法对速度参量进行修

正。在仿真的整个过程中 PA-MPA 算法的速度值都是恒定不变的，都采用初始值。PA-MPA 算法的定位误差随时间有一个增大的趋势，如果定位过程持续时间较长误差累计会更加明显。在复杂度方面，Kalman 滤波中涉及对矩阵求积、求逆运算，而 PA-MPA 算法在每个时刻计算信息时只计算均值和方差两个参数，Kalman 滤波的计算量比 PA-MPA 算法要大。

接下来我们对比粒子滤波与本文算法的定位性能。在对比这两个算法过程中，我们采用的仿真环境不变。取过程噪声方差为 10，观测噪声方差仍然为 100。在粒子滤波和 P-MPA 算法中每个时刻采样粒子数取为 1000。图 6 为用粒子滤波和本文提出的 P-MPA 算法所估计出来的目标运动轨迹。

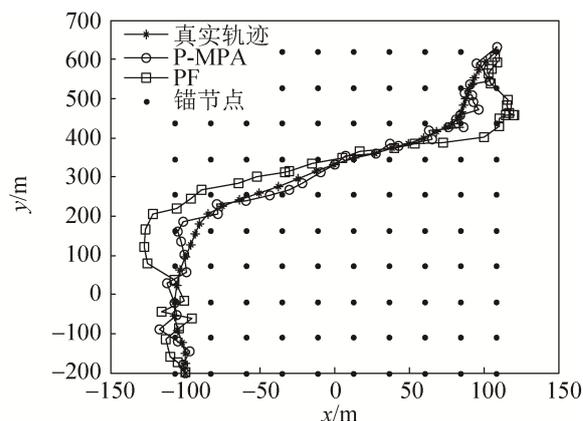


图6 P-MPA 和粒子滤波算法的定位结果

Fig. 6 The positioning results of P-MPA and particle filter algorithm

由于本文算法利用粒子的形式实现了信息传递，克服了基于参数的信息传递算法在非线性、非高斯环境中误差大的缺点。P-MPA 算法在粒子滤波的基础上运用吉布斯采样进行修正，在修正阶段 P-MPA 算法的修正信息来源于邻节点。P-MPA 算法在修正阶段参考的邻节点不仅包括邻居未知节点还包括邻居锚节点。而 PF 算法在每个时刻根据系统模型利用上一时刻的采样粒子来进行预测采样，在修正阶段利用观测模型得到的观测值进行重采样，然后求所有采样粒子的加权平均值来得到当前状态的估计值，而不参考与邻节点的实际测量

值。可以看出 P-MPA 算法的定位精度要高于粒子滤波算法, 并且两种算法的定位误差分布都比较集中, 没有随时间而增加。

最后, 图 7 为 4 种不同定位算法的误差累积分布函数(Cumulative Distribution Function, CDF)图。该过程的仿真环境不变, 取所有算法的过程噪声方差为 10, 观测噪声方差为 100。可以看出在 4 种算法中本文算法的定位精度最高, PA-MPA 的定位精度最低。

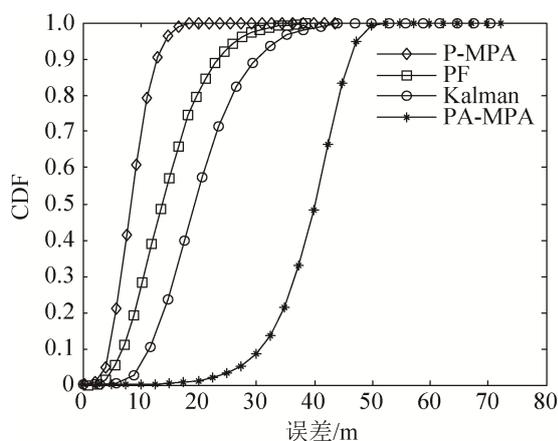


图 7 误差累积分布函数

Fig. 7 Accumulative distribution function of error

4 结论

本文将基于因子图的信息传递算法用粒子形式来实现。结合协作定位网络因子图的特点, 求因子顶点到变量顶点信息时直接利用系统模型进行重要性采样; 在求变量到因子信息时采用吉布斯采样。仿真结果表明, 粒子形式的信息传递算法的定位精度明显高于基于参数的信息传递算法。此外, 我们还与传统的卡尔曼滤波算法和粒子滤波算法做了对比, 结果表明本文算法在精度方面是最优的。本文算法在协作阶段选择的是固定通信范围内的邻节点。然而, 采用固定通信范围有可能导致由于节点密度小使得参与协作的邻节点数量较少和由于邻节点密度大使得参与协作的邻节点数量过大。因此, 邻节点选择策略有可能成为我们下一步的研究方向。

参考文献:

- [1] Tseng P H, Ding Z, Feng K T. Cooperative Self-Navigation in a Mixed LOS and NLOS Environment[J]. IEEE Transactions on Mobile Computing (S1536-1233), 2014, 13(2): 350-363.
- [2] Li S, Wang X. Joint composite detection and Bayesian estimation: A Neyman-Pearson approach[C]// IEEE Global Conference on Signal and Information Processing. 2015: 453-457.
- [3] Bar S, Tabrikian J. Bayesian Estimation in the Presence of Deterministic Nuisance Parameters—Part II: Estimation Methods[J]. IEEE Transactions on Signal Processing(S1053-587X), 2015, 63(24): 1.
- [4] Wu A G, Qian Y Y, Wu W J. Bias compensation-based recursive least-squares estimation with forgetting factors for output error moving average systems[J]. Iet Signal Processing(S1751-9675), 2014, 8(5): 483-494.
- [5] Dalton L A, Dougherty E R. Bayesian Minimum Mean-Square Error Estimation for Classification Error - Part I: Definition and the Bayesian MMSE Error Estimator for Discrete Classification[J]. IEEE Transactions on Signal Processing(S1053-587X), 2011, 59(1): 115-129.
- [6] 黄小平. 卡尔曼滤波原理及应用[M]. 北京: 电子工业出版社, 2015.
Huang Xiaoping. Calman filtering principle and application [M]. Beijing: Electronic Industry Press, 2015.
- [7] Nerurkar E D, Roumeliotis S I, Martinelli A. Distributed maximum a posteriori estimation for multi-robot cooperative localization[C]// IEEE International Conference on Robotics and Automation. IEEE, 2009: 1402-1409.
- [8] Cakmak B, Urup D N, Meyer F, et al. Cooperative Localization for Mobile Networks: A Distributed Belief Propagation - Mean Field Message Passing Algorithm[J]. IEEE Signal Processing Letters(S1070-9908), 2016, 23(6): 828-832.
- [9] Shi Y, Cui Q, Cao S, et al. Performance relationship between distributed and centralised cooperative localisations[J]. Electronics Letters(S0013-5194), 2014, 50(50): 127-128.
- [10] Wymeersch H, Lien J, Win M Z. Cooperative localization in wireless networks[J]. Proceedings of the IEEE (S0018-9219), 2009, 97(2): 427-450.
- [11] Win M Z, Conti A, Mazuelas S, et al. Network localization and navigation via cooperation[J]. IEEE Communications Magazine (S0163-6804), 2011, 49(5): 56-62.

(下转第 2917 页)

<http://www.china-simulation.com>

• 2907 •