

極限的実装によるシミュレーション・パラメータ推定用無限粒子フィルタの開発

林圭佐*, 齋藤正也*, 樋口知之**

所属名, *科学技術振興機構, CREST, **統計数理研究所

メールアドレス, ghayashi@ism.ac.jp

概要: 超並列コンピューティングに適合性が良い、スケーラビリティの良い粒子フィルタを実装した。各ノードで行う処理は非同期かつ独立であり、初期データの転送、処理結果の集計以外に通信は発生しない。非同期独立性からインクリメンタルに粒子を追加できるため、パラメータの事後確率分布が収束するまで粒子数を増加可能であり、粒子数は事実上無制限である。試算の結果、67 倍程度の高速化が得られた。

1 はじめに

データ同化は気象・海洋シミュレーションの分野で発達してきた手法である。完全ではないモデルと観測されたデータを融合することで、現実に近い計算結果を得るほか、適切な初期値・パラメータの推定にも用いられる。その有用性は広く認知され、気象・海洋の分野のみならず宇宙空間、ゲノム情報と分野を超えた広がりが見られる^[1,2]。

ゲノム情報におけるデータ同化^[3]では、常微分方程式で記述される生化学反応ネットワークと反応濃度の時系列データを利用して生化学反応ネットワークのパラメータ推定を行う場合、1 億個の超多粒子を用いた粒子フィルタが有効に働くことが確認された^[4]。超多粒子による粒子フィルタを推進するため、超並列コンピューティングに適合した粒子フィルタを開発する。

2 ベイズ推定によるパラメータ推定

次のような非線型状態空間モデルで記述した問題を考える。

システムモデル: $x_t = f(x_{t-1}) + v_t, v_t \sim p(v_t)$.

観測: $y_t = h(x_t) + w_t, w_t \sim p(w_t)$.

初期値を含めたモデルパラメータ: $\tilde{\theta} \sim p(\tilde{\theta})$.

以上で定義される状態空間モデルにおいて、与えられた観測 $\{y_t\}_{t=1,2,\dots,T}$ のもとで、初期値を含めたモデルパラメータ $\tilde{\theta}$ の事後確率分布 $p(\tilde{\theta}|y_{1:T})$ を推定する。

ベイズの定理から、 $\tilde{\theta}$ の事後確率分布は事前確率分布と尤度の積に比例する。

$$p(\tilde{\theta}|y_{1:T}) \propto p(\tilde{\theta})p(y_{1:T}|\tilde{\theta}). \quad (1)$$

式(1)の右辺は尤度の分解の公式を用いて

$$p(\tilde{\theta})p(y_{1:T}|\tilde{\theta}) = p(\tilde{\theta})\prod_{j=1}^T p(y_j|y_{1:j-1}, \tilde{\theta}), \quad (2)$$

と書くことができる。

式(2)の右辺のうち、事前確率分布 $p(\tilde{\theta})$ は研究者の先験情報から与える。一時点尤度は、観測誤差の確率分布と観測時刻 $t-1$ までの観測データを同化した予測確率分布から求まる。

$$p(y_t|y_{1:t-1}, \tilde{\theta}) = \int p(y_t|x_t, \tilde{\theta})p(x_t|y_{1:t-1}, \tilde{\theta})dx_t.$$

3 Sequential Importance Sampling

既報^[4]では、観測データを同化した確率分布を求めるために、粒子フィルタ(SIR)を用いた。粒子フィルタでは、リサンプリングにより実現値の集合でモンテカルロ近似された状態変数の分布に観測データの情報を取り込む。リサンプリングには、実現値集合と観測から求めた尤度の合計を確定する必要があり、場合により通信が発生する。このため、尤度の合計を求める縮約、粒子情報交換に伴う通信が並列計算の性能低下を招く。本報では、Sequential Importance Sampling (SIS)を用いる。SIS 型粒子フィルタではリサンプリングの代わりに、尤度に基づいた重みを用いる。SIS で同化すると、予測確率分布は

$$p(x_t|y_{1:t-1}, \tilde{\theta}) \cong \sum_{i=1}^N \omega_{t-1|t-1}^{(i)} \delta(x_t - x_{t-1}^{(i)}|t),$$
$$\omega_{t-1|t-1}^{(i)} = \frac{\omega_{t-2|t-2}^{(i)} p(y_{t-1}|x_{t-1}^{(i)}, \tilde{\theta})}{\sum_{j=1}^N \omega_{t-2|t-2}^{(j)} p(y_{t-1}|x_{t-1}^{(j)}, \tilde{\theta})}. \quad (3)$$

で近似される。ここで δ は Dirac のデルタ関数、 N はモンテカルロ近似に用いる粒子数である。

図1に $x_0 \sim p(x_0)$ で粒子を発生した場合の SIS による重みの更新状況を模式的に示す。重みの分母の計算は、全粒子にわたる縮約が必要ゆえにオーバーヘッドが大きいと、並列処理中の式(4)の分

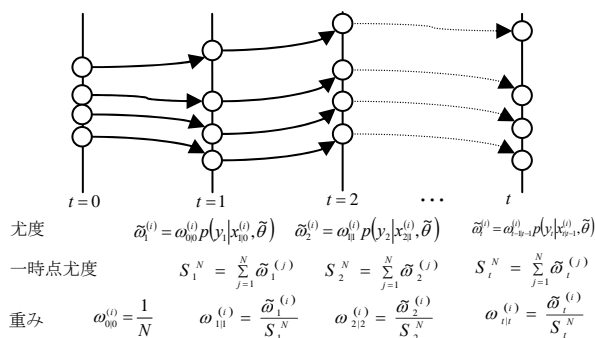


図 1. SIS 模式図

母を未確定として SIS を非同期に実装する。各観測時刻で尤度を保存し、全粒子の計算後にまとめて尤度関数の値を計算すると、式(2)が

$$p(\tilde{\theta}) \prod_{j=1}^T p(y_j | y_{1:j-1}, \tilde{\theta}) = p(\tilde{\theta}) \frac{1}{N} \sum_{i=1}^N \prod_{t=1}^T p(y_t | x_{t,t-1}^{(i)}, \tilde{\theta}) \quad (4)$$

となることが導ける。

式(4) から、事後確率分布は、観測データの条件付き確率の加重平均で表される。つまり、モンテカルロ近似に用いる粒子に関して、それぞれ独立に計算することが可能であるし、結果として粒子を逐次的に増加可能である。粒子の増加に用いるデータは、尤度 $S^N = \frac{1}{N} \sum_{i=1}^N \prod_{t=1}^T p(y_t | x_{t,t-1}^{(i)}, \tilde{\theta})$ と粒子数 N だけで良いので、省メモリであり、通信量を低く抑えることができる。 N 粒子と M 粒子による結果を合算するには次式を用いる。

$$L^{N+M}(\tilde{\theta}) = \frac{N}{N+M} L^N(\tilde{\theta}) + \frac{M}{N+M} L^M(\tilde{\theta})$$

粒子追加の処理は収束するまで粒子数を追加可能だから、このように粒子を増大して収束計算する SIS の構成を「無限 SIS」と呼ぶことにする。

以上で見たように、本手法は各粒子の計算が独立であり、通信量が少ないためスケーラビリティの高い手法であることが期待できる。

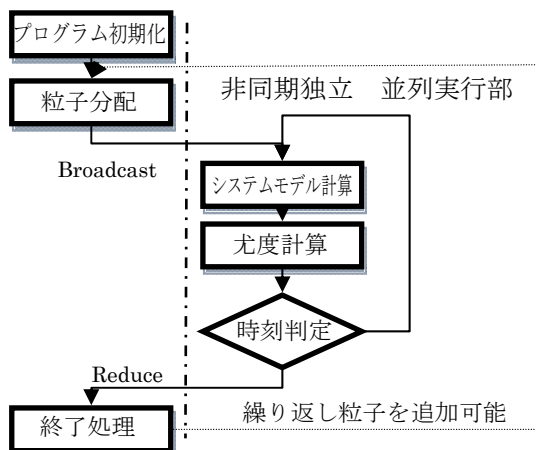


図 2. 流れ図

3 GPGPU による数値実験

文献[4]で用いられたサーカディアンリズムモデルのパラメータ推定について、AMD Opteron(tm)

Processor 2220 の PC に接続した Tesla C870 を用いて計算した。CUDA でパフォーマンスを得るためには並列に実行できる計算を GPU で行い、高速なメモリを適切に使い、ホストと GPU 間のデータ転送を少なくすることが肝要である。本手法は、各粒子の計算は独立であり、共用できるパラメータを高速な共有メモリに置き、状態変数をレジスター上に置いて計算できたため、理想的な問題だったといえる。本手法で 1 億粒子の処理にかかる時間は 3 時間程度である。計算機アーキテクチャが違うため比較が難しいが、Opteron 2220 1Core における報告との比較を図 4 に示す。

	粒子数	一億粒子処理時間	倍率	機材
無限 SIS	1×10^8	1.0×10^4	67	Tesla D870+Opteron2220, 1Core
従来法	1×10^8	6.8×10^5	1.0	Opteron2220, 1Core

図 4. 従来法との比較

4 まとめ

超並列コンピューティングでも効率よく実行可能な粒子フィルタアルゴリズムについて議論した。

従来法は観測データを同化する際、システム全体が同期する必要がある。「無限 SIS」は各粒子が同期不要で独立に計算でき、シミュレーション中には、いかなる通信も発生しない。試計算の結果、GPGPU 環境下において 67 倍速の高速化が得られた。

謝辞 この研究は明治大学 中村和幸特任講師、統計数理研究所 吉田亮助教、東京大学 医科学研究所 長崎正朗助手、宮野悟教授らから基本となるソフトウェアの提供を受けました。厚く御礼申しあげます。

参考文献

- [1] 樋口知之 (監修・著)、「統計数理は隠された未来をあらわにする：ベイジアンモデリングによる実世界イノベーション」、東京電気大学出版局、2007.
- [2] 樋口知之、「全体モデルから局所モデルへ/状態空間モデルとシミュレーション」、数学セミナーII、Vol. 46, No. 11, 30-36, 2007.
- [3] Nagasaki, M. et al.; Genomic Data Assimilation for Estimating Hybrid Functional Petri Net from Time-course Gene Expression Data, Genome Informatics, Vol. 17, 46-61, 2006.
- [4] Nakamura, K. et al.; Parameter estimation of in silico biological pathways with particle filtering towards a petascale computing, Pacific Symposium on Biocomputing. 14: 227-238, 2009.