情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2009-MPS-76 No.19
Vol.2009-BIO-19 No.19
2009/12/17

GPGPUを活用した バイオロジカルパスウェイモデルの高速なパラメータ推定

林　圭佐[†1,†2]　齋藤正也[†1,†2]
吉田　亮[†2]　樋口知之[†2]

バイオロジカルパスウェイモデルのパラメータ推定は探査空間が高次元空間となり、モデルの非線形性もあいまって困難な問題となりうる。近年話題に上ることが多い GPGPU の並列計算機能を活かした粒子フィルタを実装した。SIS 粒子フィルタを用いて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。試計算の結果、67 倍の高速化が得られた。

A parameter estimation for Biological pathway model using GPGPU

Keisuke HAYASHI,[†1,†2] Masaya SAITO,[†1,†2]
Ryo YOSHIDA[†2] and Tomoyuki HIGUCHI [†2]

The parameter estimation of biological pathway models is a difficult problem. The difficulty comes from the high dimensionality and the nonlinearity make of models. We combat this problem by means of an parallelised particle filter algorithm dedicated to parallel environment, particular GPGPU. Our algorithm is based on SIS (Sequential Importance Sampling) and hence no communication occurs during the simulation. We have increased the computation speed 67 times, compared to a computation with a single CPU core.

†1 科学技術振興機構/CREST
JST/CREST
†2 統計数理研究所
The Institute of Statistical Mathematics

1. はじめに

近年、描画処理ユニットの目的外使用の試みが盛んに行われている。いくつかの分野では GPU を適切に用いることで飛躍的な速度向上がなされている[5]。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、モデルと観測されたデータを融合する手法である。データに基づき現実に近づけた計算結果を得るほかにも、適切な初期値・パラメータの推定にも用いられている。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)]。本報では超多粒子による粒子フィルタを推進するため、データ同化手法の一つ粒子フィルタを GPGPU 向けに実装したので報告する。

2. パラメータのベイズ推定

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。

$$\begin{aligned}
&\text{システムモデル：} && x_t = f(x_{t-1}) + v_t, \quad v_t \sim p(v_t)\\
&\text{観測モデル：} && y_t = h(x_t) + w_t, \quad w_t \sim p(w_t)\\
&\text{パラメータの事前分布：} && \tilde{\theta} \sim p\left(\tilde{\theta}\right)
\end{aligned}$$

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 $\{y_t\}_{t=1,2,\ldots,T}$ 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 $p\left(\tilde{\theta}\middle| y_{1:T}\right)$ 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。

$$p\left(\tilde{\theta}\middle| y_{1:T}\right) \propto p\left(\tilde{\theta}\right) p\left(y_{1:T}\middle|\tilde{\theta}\right) \tag{1}$$

式 (1) の右辺は尤度の分解の公式を用いて

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

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 までの観測データを同化した予測確率分布から求まる。

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

Vol.2009-MPS-76 No.19
Vol.2009-BIO-19 No.19
2009/12/17

3. 計 算 手 法

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) を用いている。粒子フィルタでは、状態変数の分布をモンテカルロ近似により、粒子と呼ばれる実現値の集合で表現する。そして、観測データの情報を融合したシミュレーション結果を得るため、リサンプリングを実施する。リサンプリングは、尤度をもとに計算した確率で粒子を選択抽出する。

GPGPU のプログラミングモデルは、用意された大量の演算器を有効活用するための、多数のスレッドに分けられた演算が同時に実行するようになっている。本報で用いた GPU でパフォーマンスを出すためは、メモリアクセスが局所化されるようにし、大域メモリへのロードストアが連続番地となっているようにプログラムを調整することが要求される。

リサンプリングを素直に実装した場合、尤度の合計を求めるためのスレッド間通信、リサンプリングにおける不揃いな分岐、粒子の情報交換に伴うメモリへのランダムアクセスが GPGPU の実行性能低下を招いた。

これらの問題を解消するために、本報は Sequential Importance Sampling (SIS) を採用した。SIS 型粒子フィルタではリサンプリングの代わりに、尤度に基づいた粒子の重みを用いる。SIS では、予測確率分布は

$$p\left(x_t \left| y_{1:t-1}, \tilde{\theta}\right.\right) \simeq \sum_{i=1}^{N} \omega_{t-1|t-1}^{(i)} \delta(x_t - x_{t-1|t}^{(i)}),$$

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

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。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ため、並列処理中の式 (3) の分母を確定せずに SIS を非同期に実行する。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場合、式 (2) が

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

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 $L^N = \frac{1}{N} \sum_{j=1}^{N} \prod_{t=1}^{T} p\left(y_t \left| x_{1:t}^{(i)} y_{1:t-1}, \tilde{\theta}\right.\right)$ 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。図 1 に、$L^N(\theta)$ を計算する手続きを示す。

$$L^{M+N}\left(\tilde{\theta}\right) = \frac{ML^M\left(\tilde{\theta}\right)}{M+N} + \frac{NL^N\left(\tilde{\theta}\right)}{M+N}.$$

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。

(a) 全体　　(b) 尤度計算部

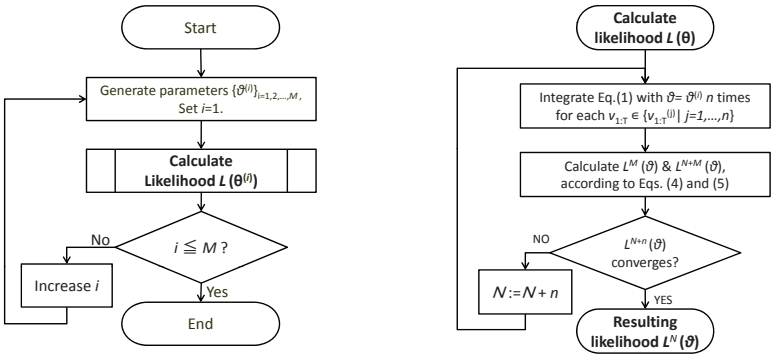

図 1 逐次的な粒子追加により尤度の精度を改良する手続き

4. GPGPU による数値実験

図 2 に示したサーカディアンリズムモデルにおいて、パラメータ推定を AMD Opteron(tm) Processor 2220 を搭載した PC に接続した Tesla C870 を用いて行った。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

Vol.2009-MPS-76 No.19
Vol.2009-BIO-19 No.19
2009/12/17

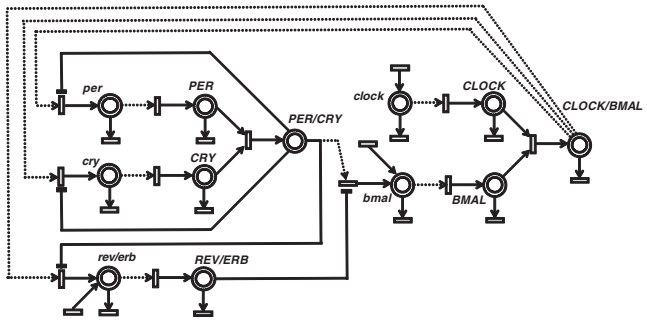

図 2 サーカディアンリズムモデルの HPFN による表現 (既報[6] の Fig. 4 を再掲).

表 1 計算時間

	粒子数	処理時間 (sec)	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

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[4] との比較を表 1 に示す。

5. おわりに

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, 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, pp.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, pp.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: pp.227-238,(2009).
5) Cuda Zone `http://www.nvidia.co.jp/object/cuda_home_jp.html`
6) R. Yoshida, M. Nagasaki, R. Yamaguchi, R. Imoto, S. Miyano, and T. Higuchi. Bayesian learning of biological pathways on genomic data assimilation. *Bioinformatics*, 24:pp.2592–2601, (2008).