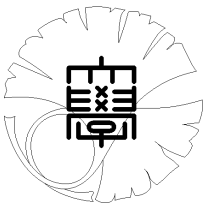
数理科学実践研究レター　2023–6　March 30, 2023

数値シミュレーションモデルの効率的学習に向けた適応サンプリング手法の検討

by

安藤 大悟、平松 遼太



UNIVERSITY OF TOKYO

GRADUATE SCHOOL OF MATHEMATICAL SCIENCES

KOMABA, TOKYO, JAPAN

数値シミュレーションモデルの効率的学習に向けた適応サンプリング手法の検討

安藤大悟[1]（東京大学大気海洋研究所）
Taigo Ando (Atmosphere and Ocean Research Institute, The University of Tokyo)

平松遼太（ダイキン工業株式会社）
Ryota Hiramatsu (DAIKIN INDUSTRIES, LTD.)

概 要

本研究では、シミュレーションモデルの結果を機械学習モデルに学習させる場合を念頭に、適応サンプリング手法を利用することで応答曲面モデルの学習速度がどのように変化するのかを調べた。入力変数が多い場合や非線形性があまり大きくない場合には適応サンプリングがあまり有効ではないことが示されたが、手法の改良によってサンプリングの効率が向上する可能性が示唆された。

1 はじめに

機械学習技術の発達と普及にともなって、産業界においてもいかに機械学習を活用するかが大きな課題となっている。そうした活用例の 1 つとして、製品の開発や設計に用いる複雑な物理モデルを応答曲面モデルに学習させ代替とすることで計算コストを削減することが挙げられる。この場合、効率的に学習を進めるためには、どのように元のモデルから教師データを生成するかという実験計画法を検討することが重要である。一つの方針は、条件領域をできるだけ均質にカバーするようにサンプル点を取るというものである。一方、サンプル点を少しずつ増やしながら元のモデルがどのような条件で非線形的な応答を示しやすいかを評価し、そのような領域に重点的にサンプル点を足していくという方針も考えられ、このような手法は適応サンプリングと呼ばれる。本研究では、実際の業務で用いられるシミュレーションモデルに近いようなテストケースを用いて、適応サンプリングに基づく学習の効率化の可能性を検討する。

2 モデルの構成

2.1 サンプリング手法 (cf.[1])

本研究では、[1] によって提案された Lipschitz Sampling のアルゴリズムに概ね従ってサンプリングを行う。以下では Euclid 空間上で定義される未知関数 f を学習するためにサンプリング点を選択することを考える。Lipschitz Sampling では、サンプリングの候補点 $\boldsymbol{x}$ を既存のサンプリング点集合 $\mathcal{X}$ から計算される以下のメリット関数によって評価する。

$$merit(\boldsymbol{x}) = L(\boldsymbol{x}) \times Radius(\boldsymbol{x})$$

ここで $L(\boldsymbol{x})$ は候補点付近における Lipschitz constant の推定値であり、これは最も近い既存サンプリング点 $\boldsymbol{x^i} \in \mathcal{X}$ を用いて以下のように与える。

$$L(\boldsymbol{x}) = \sup_{\substack{\boldsymbol{x^j} \in \mathcal{X}_{adj} \\ i \neq j}} \frac{\left| f(\boldsymbol{x^i}) - f(\boldsymbol{x^j}) \right|}{\left\| \boldsymbol{x^i} - \boldsymbol{x^j} \right\|}$$

$\mathcal{X}_{adj}$ は $\boldsymbol{x^i}$ 周囲のサンプリング点の集合であり、$\boldsymbol{x^i}$ と $\boldsymbol{x^j} \in \mathcal{X}_{adj}$ は既存サンプリング点による Voronoi 分割を行ったときに領域が隣接する。
また、$Radius(\boldsymbol{x})$ は最も近い既存サンプリング点との距離

$$Radius(\boldsymbol{x}) = \min_{\boldsymbol{x^i} \in \mathcal{X}} \left\| \boldsymbol{x} - \boldsymbol{x^i} \right\|$$

[1] t-ando@aori.u-tokyo.ac.jp

である。
具体的なサンプリング手順は以下の通りである。まず、モデルの入力変数の範囲から空間を均質に充填するようなサンプリング手法として知られる Latin Hypercube Design によって初期サンプリング点を設定する (本研究では 50 個を設定)。続いてサンプリング点によって変数領域を Voronoi 分割し、その頂点を追加するサンプリング点の候補とする。これらの候補点に対して上述したようにメリット関数を計算し、値の大きいものから順にいくつかを追加サンプリング点とする。以降は新たにサンプリングされた点を加えて Voronoi 分割から繰り返し、目標とするサンプリング数に達するまで続けられる。
Lipschitz Sampling には様々なパラメーターや調整可能な点が存在するが、特に 1 ステップで追加するサンプリング点の数は結果に大きな影響を与える。計算コストの面からは一度に大量のサンプリング点を追加する方が望ましいが、その場合は変数空間内で偏りが生じやすくなるため学習効率が落ちると考えられる。そこで、サンプリング点の追加について以下の 2 パターンを実施した。

Lip_A: 候補点のうち半分をサンプリング点として追加する。これは [1] と同様の手法である。

Lip_B: 候補点のうち最大半分をサンプリング点として追加するが、1 ステップでの追加数に上限を設ける。さらに、同一ステップでの追加サンプリング点が既存の点から一定距離 r_{crit} 以内にある場合、その候補点を追加しない。

Lip_B における追加数上限 N_{addmax} や r_{crit} はチューニングパラメーターである。以下では $N_{addmax} = 100$ とした。また、r_{crit} は変数空間の次元 d や既存サンプリング点の個数 N に依存させるのが望ましいと考えられるが、本研究では変数領域に対して半径 r_{crit} の球が占める領域が大きくない d に対してあまり変わらないように

$$r_{crit} = r_0 \frac{\log(d+2)}{\sqrt[d]{N}}$$

とした。以下では、サンプリング領域を $x_i \in [0,1]$ とする規格化に対し $r_0 = 0.05$ とした結果を示す。

2.2 モデル評価

各サンプリング手法を以下の 2 種類のテスト関数に用いて評価する。

1) doubly-bumped function

$$f(\boldsymbol{x}) = \exp\left(-5\frac{\sum_{i=1}^{d} {x_i}^2}{2}\right) + 2\exp\left(-100\frac{\sum_{i=1}^{d}(x_i - 0.6)^2}{2}\right), \qquad x_i \in [-1,1]\ (i = 1,2,\cdots,d)$$

2) evaporator-like function

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} i \cdot \exp({x_i}^2), \qquad x_i \in [0,1]\ (i = 1,2,\cdots,d)$$

ここで d は次元数を表す。後者はサンプリング領域の端で勾配が大きくなるような特徴を持つが、これは実際の製品開発に用いられる数値モデルの出力の特徴と類似したものとなっている。以下では $d = 2$ および $d = 4$ のケースについて評価する。
学習モデルには勾配ブースティング手法である LightGBM[2] を使用する。各テストケースに対して Latin Hypercube Design (LHD)、Lip_A、Lip_B のそれぞれで最大 1000 点までサンプリング点を取得する。それぞれ 10 回のサンプリングを行い、R2 スコアの平均値で各モデルを評価する。これは、テストデータの標本値 $\boldsymbol{y} = \{y_1, y_2, \cdots, y_N\}$ に対する予測値 $\boldsymbol{e} = \{e_1, e_2, , \cdots, e_N\}$ を得たとき、$\boldsymbol{y}$ の平均値 $\bar{y}$ を用いて

$$R^2 \equiv 1 - \frac{\sum_{i=1}^{N}(y_i - e_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

として定義される。

3　結果

3.1　2 次元の場合

まずは 2 次元の関数を用いてベンチマークテストを行った。図 1 に関数の概形および Lip_A によるサンプリングの例を示す。doubly-bumped function では山の部分、evaporator-like function では領域の端に近い部分にそれぞれサンプリングが集中しており、勾配の大きい領域を集中的にサンプリングする Lipschitz Sampling の効果が確認できる。

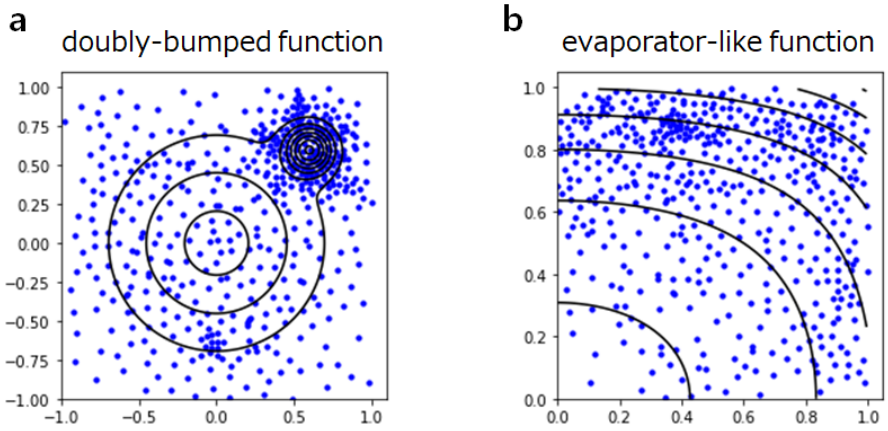


図 1: 2 次元テスト関数の概形 (コンター) および Lipschitz Sampling によるサンプリング例 (青点)。サンプリング数 500。(a) doubly-bumped function, (b) evaporator-like function

図 2 は各サンプリング手法における学習効率の比較である。doubly-bumped function については、LHD よりも Lipschitz Sampling の方がよい学習効率を得られている。一方、evaporator-like function の場合は LHD と Lipschitz Sampling で同程度の性能であった。いずれの関数でも Lip_B と Lip_A の間に有意な差は見られなかった。両関数を比較するとどのサンプリングでも evaporator-like function の方が学習速度が速く、これは doubly-bumped function よりも evaporator-like function の方が非線形性が小さいことから理解できる。また、Lipschitz Sampling によるサンプリング点の集中は特に doubly-bumped function のような非線形性の大きい関数に対して有効に働くと考えられる。

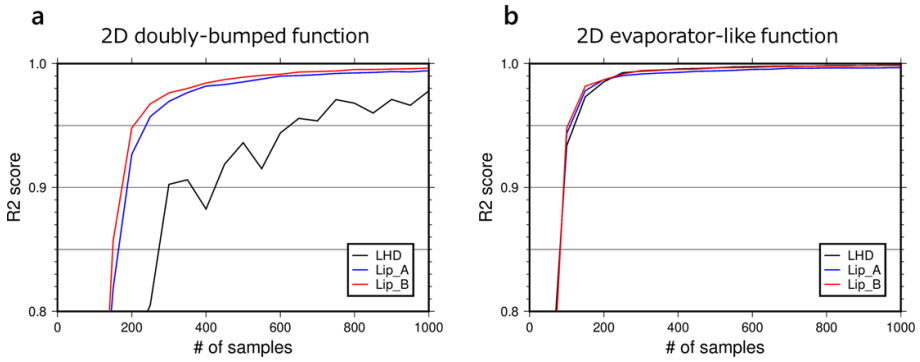


図 2: 各サンプリング手法を用いた場合のサンプリング数に対する R2 スコアの変化。テスト関数は 2 次元の (a) doubly-bumped function, (b) evaporator-like function

3.2　4 次元の場合

図 3 は 4 次元の関数に対する学習効率を比較した結果である。2 次元の場合と比べると、次元が上昇したことによりすべてのケースで学習速度が遅くなっている。その中でも特に Lipschitz Sampling

を用いた場合の効率悪化が著しく、Lip_A については doubly-bumped function と evaporator-like function の両方で LHD よりも悪い結果となった。Lip_B については Lip_A よりも有意に学習効率が上昇しており、両関数に対しても LHD と同程度の性能であった。この結果は、特に次元数の大きい場合について Lipschitz Sampling におけるサンプリング点の偏りが性能の悪化に働きやすくなる危険性を示しているが、一方で手法やパラメーターの調整によりそれが低減できるという可能性も示唆している。

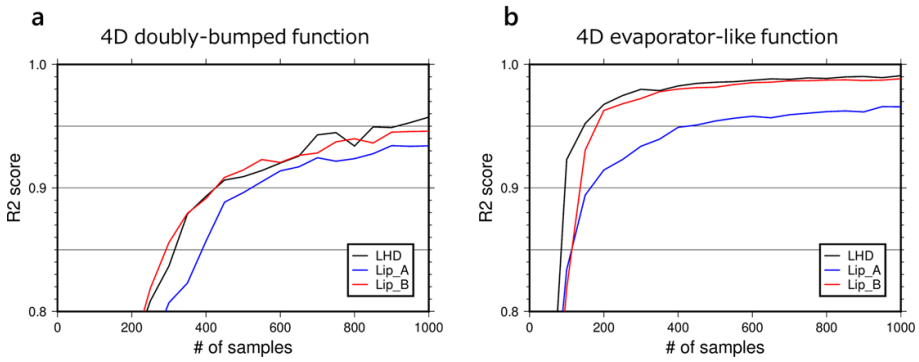


図 3: 図 3.2 と同じだがテスト関数が 4 次元の場合

4　終わりに

本研究では、Lipschitz Sampling による機械学習の効率化の可能性を調べるためいくつかの具体的なテスト関数に対してサンプリングを行った。Lipschitz Sampling は非線形性の大きい関数に対しては学習の高速化に有効に働くものの、より実用に近い次元数が大きい場合や非線形性があまり大きくない場合についてはあまり効果を発揮できず、むしろ学習効率が落ちる場合もあることが示された。一方で、不均一性を低減する工夫によって効率を改善できる可能性も示唆された。

Lipschitz Sampling のような適応サンプリングを実用する上では、さらに今後検討すべき課題が存在する。実際のシミュレーションモデルでは一般に複数の出力変数が存在するため、手法の拡張が必要となる。また、本研究で導入したものも含めて様々なパラメーターが存在しており、効率化に向けてそれらをどう適切にチューニングしていくかも大きな課題である。さらに、実用上のサンプリングの効率を判断するためには、シミュレーションモデルの計算コストとサンプリング手法自体の計算コストを並列化の度合いなど実際の計算資源状況も加味して比較する必要がある。

5　謝辞

ダイキン工業株式会社の布川賢一様、田中昌弘様、BADHAN Pragun 様、奥井隆宗様、横瀬清識様、高根沢悟様には、課題の提供および技術的支援をいただきました。また、東京大学大学院数理科学研究科の儀我美一様、間瀬崇史様や、同じ空調班として課題に取り組んだ FMSP コース生の榎優一様、佐藤翔一様、ならびに前述のダイキン工業株式会社の皆様には議論を通して多くの知見をいただきました。ここに感謝の意を表します。本研究は FMSP リーディング大学院の助成を受けております。

参考文献

[1] Lovison, A., Rigoni, E. (2011). Adaptive sampling with a Lipschitz criterion for accurate metamodeling. *Communications in Applied and Industrial Mathematics, 1*(2), 110-126.

[2] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.