Constructing quadrature rules with sparse nodes by improved kernel herding methods

（カーネルハーディング法の改良による疎な標本点をもつ求積公式の構成）

数理情報学専攻 48206223　辻和真
指導教員　田中健一郎 准教授

1　序論

本研究では以下のような再生核 Hilbert 空間 (RKHS) 上の求積公式であるカーネル求積を考える:

$$\int_\Omega f(x)\mu(\mathrm{d}x) \approx \sum_{i=1}^n \omega_i f(x_i) \quad (\Omega \subset \mathbb{R}^d, f \in \mathcal{H}_K).$$

ここで $\mathcal{H}_K$ は RKHS で, μ は Ω 上の確率測度である. Kernel Herding[1, 3, 7] はカーネル求積公式の構成法の一種である. アルゴリズムの計算的効率性や出力される数値積分公式の数値的安定性などの特長がある一方で, 標本点数に対する誤差の収束速度が他のカーネル求積法と比較して遅いという欠点がある. **本研究ではその特長を保ちつつ, 少ない標本点数で高い精度を持つ求積公式を出力するように Kernel Herding を改良する.**

2　Kernel Herding と CG 法

Kernel Herding は有限次元の凸最適化手法である条件付き勾配法 (CG 法 [6]) の無限次元の特別な場合と考えられる. CG 法で解くのは次の問題:

$$\min_{x\in C} f(x) \quad (f: \text{凸}, C \subset \mathbb{R}^d: \text{凸多面体}).$$

ここで C に対して $C = \mathrm{conv}(V_C)$ $(V_C \subset C)$ となるとする. アルゴリズムは以下の通り:

Algorithm　条件付き勾配法 (CG 法)

Require: 初期点 $\xi_1 \in V_C$

1: **for** $t = 1$ to $n-1$ **do**
2: 　$v_{t+1} = \mathrm{argmax}_{v\in V_C} \langle -\nabla f(\xi_t), v - \xi_t\rangle$
3: 　ステップ幅 $0 < \alpha_t \le 1$ を決める
4: 　$\xi_{t+1} = (1-\alpha_t)\xi_t + \alpha_t v_{t+1}$
5: **end for**
6: **return** ξ_n

Kernel Herding は以下の設定に対応する:

- $F(\nu) = \|\mu_K - \nu\|_K^2$
  (最悪積分誤差 (MMD) の 2 乗).
- $V_C = \{K(x,\cdot) \mid x \in \Omega\}$.
- $C = \overline{\mathrm{conv}\{K(x,\cdot) \mid x \in \Omega\}}$.

ここで $\|\cdot\|_K$ は $\mathcal{H}_K$ のノルムで $\mu_K = \int_\Omega K(x,\cdot)\mu(\mathrm{d}x)$. 出力 ν_n は $\nu_n = \sum_{i=1}^n \omega_i K(x_i,\cdot)$ になり, 標本点 $\{x_i\}_{i=1}^n$, 重み $\{\omega_i\}_{i=1}^n$ の数値積分公式に対応する. 本研究ではより小さい n で高精度な数値積分公式になるように Kernel Herding を改良する.

3　勾配近似による Kernel Herding の加速

先行研究 [4] のアルゴリズムを Kernel Herding に適用することでスパースな解を得ることを目指す.

Algorithm　勾配近似による改善版 Kernel Herding

for $t = 1$ to $t = T$ **do**
　for $k = 1$ to $k = K_t$ **do**
　　$-\nabla f(\xi_t)$ の近似勾配 $d_k = \sum_{i=1}^k c_i(K(x_i,\cdot) - \nu_t)$ $(c_i \ge 0)$ を計算
　end for
　$\nu_{t+1} = \nu_t + \alpha_t d_{K_t}$ (K_t 個の頂点が追加)
end for

近似勾配 d_{K_i} と真の勾配 $-\nabla F(\nu_i)$ のなす角を θ_i とする. 以下の命題から最悪積分誤差は $\cos\theta_i$ によって決まることがわかる.

Proposition 3.2.2. $\epsilon_i = \|\mu_K - \nu_i\|_K^2$ とする. $\alpha_i \ne 1$ $(i = 1,\dots,n)$ なら

$$\epsilon_t = \epsilon_t \cdot \prod_{i=1}^{t-1}(1 - \cos^2\theta_i).$$

これに基づき, 近似勾配の構成手法として以下のような $\cos\theta$ を貪欲に最大化する手法を提案する.

cos 貪欲最大化法

1. $c_k, v_k \leftarrow \mathrm{argmax}_{\substack{v\in V-\nu_t \\ c\ge 0}} \frac{\langle -\nabla F(\nu_t), d_k + cv\rangle_K}{\|-\nabla F(\nu_t)\|_K \|d_k + cv\|_K}$
2. $d_{k+1} = d_k + c_k v_k$

以下の定理で近似勾配 d_k の真の勾配への方向としての収束が保証される.

Theorem 3.2.2 (cos 貪欲最大化法による近似勾配の収束). $P(d_k)$ を $-\nabla F(\nu_t)$ の直線 $\{\alpha d_k \mid \alpha \in \mathbb{R}\}$ への射影とする. cos 貪欲最大化法による近似勾配 d_k に関して

$$\| -\nabla F(\nu_t) - P(d_k)\|_K^2 = O(1/k) \quad (k \to \infty).$$

また, 勾配近似の錐結合係数 $\{c_i\}_{i=1}^k$ を各反復で最適化する手法も提案した. **数値実験で提案手法がスパース性に関して顕著な改善をすることを確かめた** (Figure 1の左図). カーネルは Matérn 3/2, $\Omega = [-1,1]^2$, 分布は一様分布, 最適レートは $n^{-\frac{5}{4}}$.

4 Blended Pairwise Conditional Gradients

次に BCG 法 [5] と PCG 法 [2] という手法を組み合わせた Blended Pairwise Conditional Gradients (BPCG) という手法を提案する. この手法は有限次元の凸最適化にも Kernel Herding の場合にも適用可能.

Algorithm Blended Pairwise Conditional Gradients (BPCG)

for $t = 0$ to $T-1$ **do**
  $a_t \leftarrow \operatorname{argmax}_{v \in S_t} \langle \nabla f(\xi_t), v\rangle$ {away vertex}
  $s_t \leftarrow \operatorname{argmin}_{v \in S_t} \langle \nabla f(\xi_t), v\rangle$ {local FW}
  (S_t は ξ_t の凸結合を構成する頂点集合)
  $w_t \leftarrow \operatorname{argmin}_{v \in V(P)} \langle \nabla f(\xi_t), v\rangle$ {global FW}
  if $\langle \nabla f(\xi_t), a_t - s_t\rangle \geq \langle \nabla f(\xi_t), \xi_t - w_t\rangle$ **then**
    local pairwise 方向 $s_t - a_t$ に進む (local な更新)
  else
    FW 方向 $w_t - \xi_t$ に進み頂点を追加: $S_{t+1} \leftarrow S_t \cup \{w_t\}$ (global な更新)
  end if
end for
return ξ_T

BCG と PCG は実行可能領域が無限次元の場合 (特に Kernel Herding の場合) は収束が保証されていないが **BPCG は無限次元でも収束が保証される**. 以下の定理の Case(B) が Kernel Herding の場合の収束保証.

Theorem 4.3.1, 4.3.2. C を直径 D の凸な制約領域とする. また $\{\xi_i\}_{i=0}^T \subset C$ を BPCG の出力とする.

Case (A) f を L-平滑な凸関数として μ-強凸性を満たし, C が有限次元凸多面体ならば,

$$f(\xi_T) - f(\xi_*) = O\left(\exp\left(-cT\right)\right) \quad (T \to \infty)$$

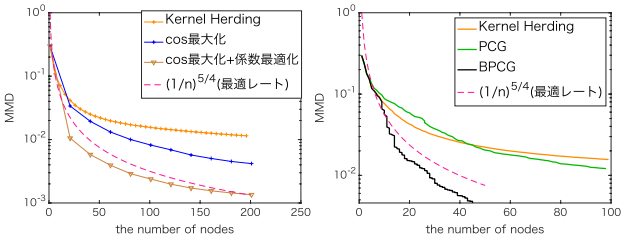

Figure 1: 3 章の実験 (左) と 4 章の実験 (右)

ここで定数 $c > 0$ は T に依存しない定数.

Case (B) f が L 平滑かつ凸な関数とすると,

$$f(\xi_T) - f(\xi_*) = O\left(\frac{1}{T}\right) \quad (T \to \infty).$$

数値実験で提案手法のスパース性に対する顕著な有効性を確かめた (Figure 1の右図). 実験条件は 3 章のものと同じ.

5 重みを最適化したカーネル求積の収束解析

$$\omega_1^*, \ldots, \omega_n^* = \operatorname*{argmin}_{\substack{\omega_i \geq 0 \\ \sum_{i=1}^n \omega_i = 1}} \left\| \mu_K - \sum_{i=1}^n \omega_i K(x_i, \cdot) \right\|_K \quad (*)$$

3 章, 4 章での提案手法に関連する $(*)$ で重みを最適化したカーネル求積公式の理論解析を考え, 関数補間との関係を利用し, **従来なされなかったカーネル依存の解析および $O(1/\sqrt{n})$ のレートよりも速い収束速度を示した** (Theorem 5.2.1).

参考文献

[1] Bach, F., Lacoste-Julien, S., and Obozinski, G. (2012). On the equivalence between herding and conditional gradient algorithms. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1355–1362, Madison, WI, USA. Omnipress.

[2] Braun, G., Pokutta, S., Tu, D., and Wright, S. (2019). Blended conditional gradients: the unconditioning of conditional gradients. In *Proceedings of the 36th International Conference on Machine Learning (PMLR)*, volume 97, pages 735–743.

[3] Chen, Y., Welling, M., and Smola, A. (2010). Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, pages 109–116, Arlington, Virginia, USA. AUAI Press.

[4] Combettes, C. and Pokutta, S. (2020). Boosting frank-Wolfe by chasing gradients. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2111–2121. PMLR.

[5] Lacoste-Julien, S. and Jaggi, M. (2015). On the global linear convergence of Frank-Wolfe optimization variants. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 496–504. Curran Associates, Inc.

[6] Levitin, E. S. and Polyak, B. T. (1966). Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50.

[7] Welling, M. (2009). Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128.