MM アルゴリズムによる行列式点過程の学習

川島 貴大[1]，日野英逸 [23]

IBISML@OIST　June 29, 2023

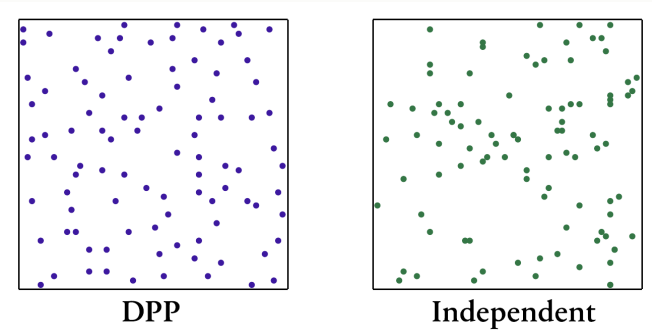
[1] 総合研究大学院大学 統計科学専攻
[2] 統計数理研究所
[3] 理研 AIP

目次

背景

ランダム部分集合の確率モデル：動機

全体集合からの多様なアイテムの生起を確率的にモデル化したい



格子点上の超一様サンプル

検索クエリからの多様な画像サジェスト

共起しやすい／しにくいアイテムの組合せを考慮できればよい

画像はそれぞれ [1] より引用.

行列式点過程

行列式点過程 (DPP; Determinantal Point Processes)

有限集合 $\mathcal{Y} = \{1, 2, \dots, N\}$ とその部分集合 $\mathcal{A} \subseteq \mathcal{Y}$ に対し，
行列式点過程が定める尤度関数は

$$P(\mathcal{A}|\boldsymbol{L}) = \frac{\det([\boldsymbol{L}]_{\mathcal{A}})}{\det(\boldsymbol{L} + \boldsymbol{I})}$$

と表される．

ここで $\boldsymbol{L} \in \mathbb{S}_+^N$ について，$[\boldsymbol{L}]_{\mathcal{A}} = (L_{ij})_{i,j \in \mathcal{A}} \in \mathbb{S}_+^{|\mathcal{A}|}$ は $\mathcal{A}$ の要素によって定められる $\boldsymbol{L}$ の主部分行列．

- 行列式によってアイテム間の共起・斥力が表現される

行列式点過程の学習

DPP の学習法はパラメータ $\boldsymbol{L}$ の構造により 3 種類に大別

- フルランク
  - EM [2]，不動点アルゴリズム [3]
- 低ランク
  - 勾配ベース最適化 [4, 5]
- その他の構造
  - クロネッカー積に分解 [6]，対角 + 特殊な低ランク構造 [7]

計算量削減のため低ランク性や特殊な構造が $\boldsymbol{L}$ に仮定される

行列式点過程の学習：フルランク

小中規模の問題はできれば L に余計な制約を与えず済ませたい

例：DPP の適用可能性を調べる予備実験

- フルランク DPP の学習
  - EM [Gillenwater et al., 2014]
    - Stiefel 多様体上での最適化が必要．複雑で不安定
  - 不動点アルゴリズム [Mariet and Sra, 2015]
    - シンプルだがハイパラを動かし加速させると収束が無保証

- 安定・高速・シンプルな DPP の学習則を考える

提案手法

平均対数尤度と方針

平均対数尤度

$$f(\boldsymbol{L}) = \frac{1}{M}\sum_{m=1}^{M} \log\det([\boldsymbol{L}]_{\mathcal{A}_m}) - \log\det(\boldsymbol{L}+\boldsymbol{I})$$

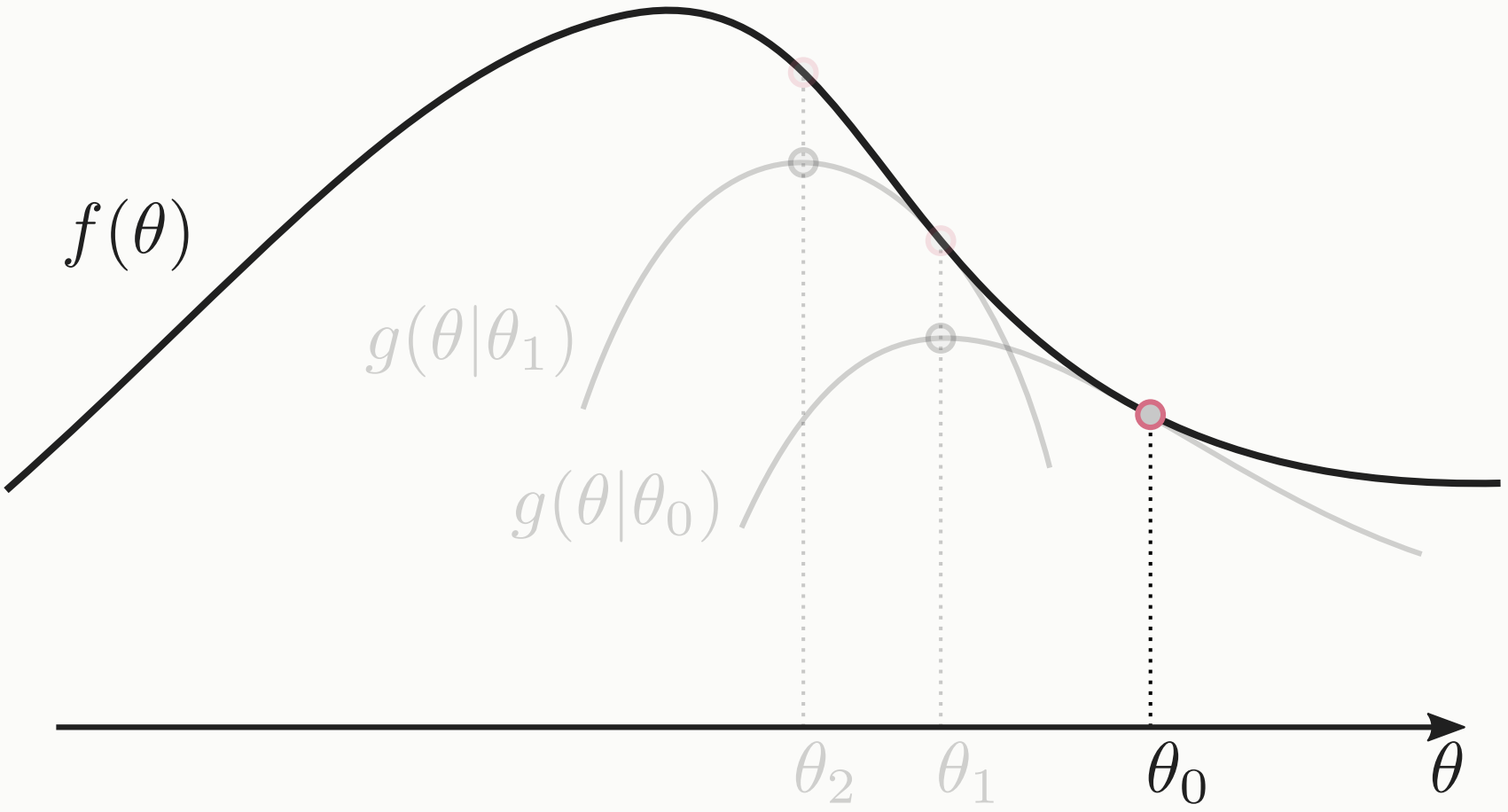
を MM アルゴリズムで最大化．つまり任意の $\boldsymbol{L}, \boldsymbol{L}^{(t)} \succeq 0$ に対し

- $f(\boldsymbol{L}) \geq g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$
- $f(\boldsymbol{L}^{(t)}) = g(\boldsymbol{L}^{(t)}|\boldsymbol{L}^{(t)})$

が成り立つ minorizer $g(\cdot|\boldsymbol{L}^{(t)})$ を設計して代理関数とし，繰り返し最大化を行う：

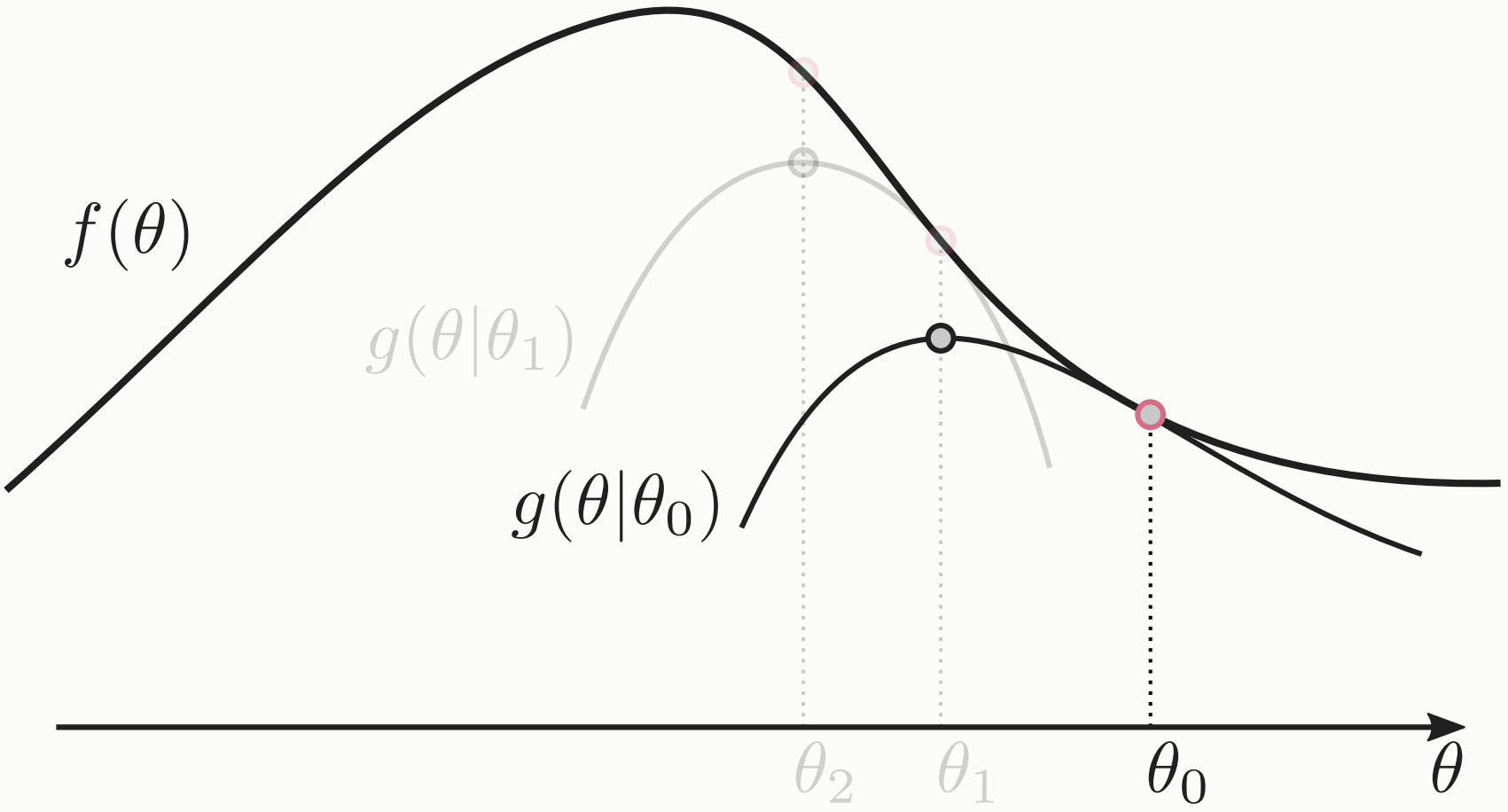
$$\boldsymbol{L}^{(t+1)} = \operatorname*{arg\,max}_{\boldsymbol{L}\succeq 0} g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$$

MM アルゴリズム



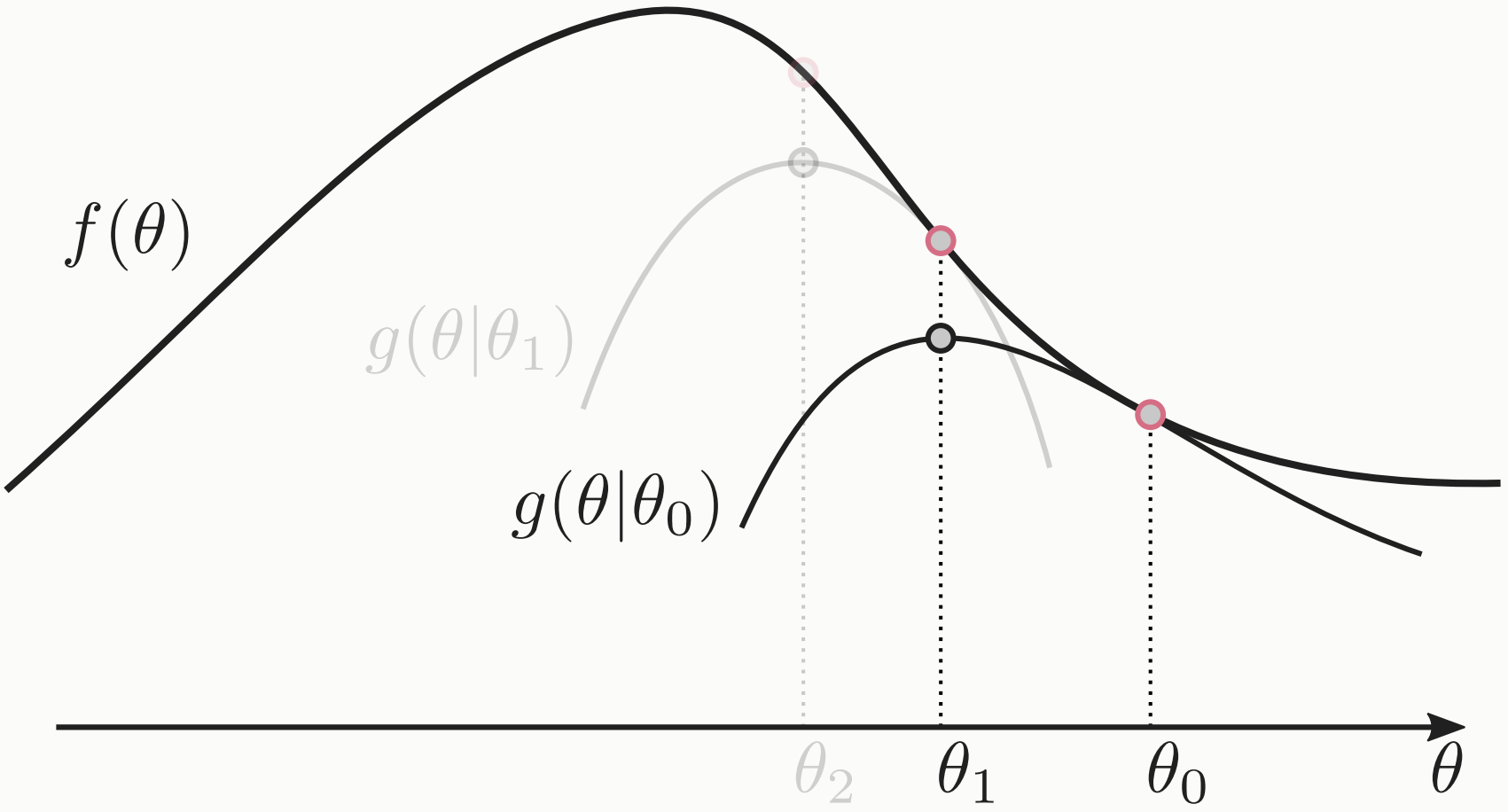
MM アルゴリズムの概要図

MM アルゴリズム



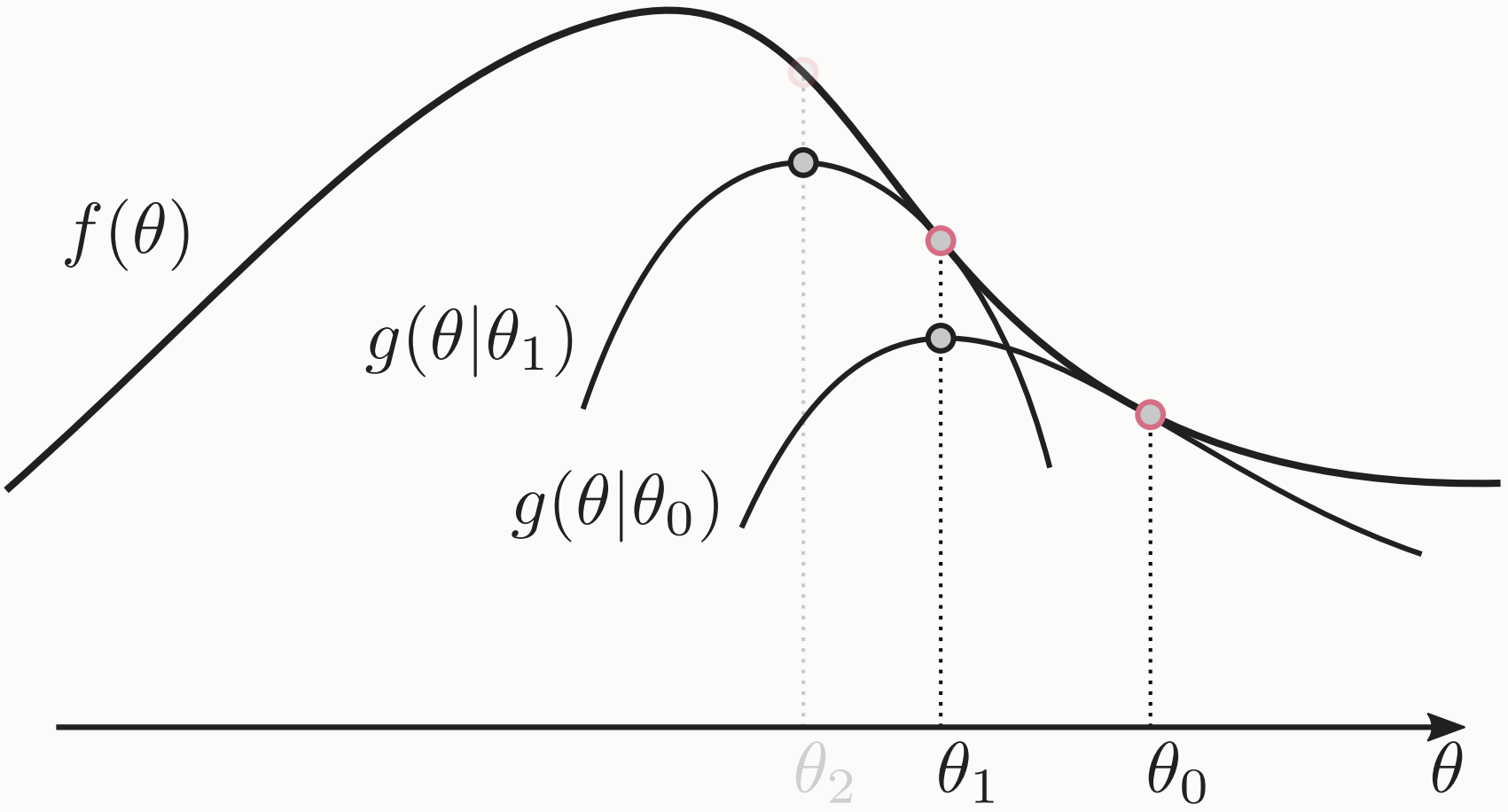
MM アルゴリズムの概要図

MM アルゴリズム



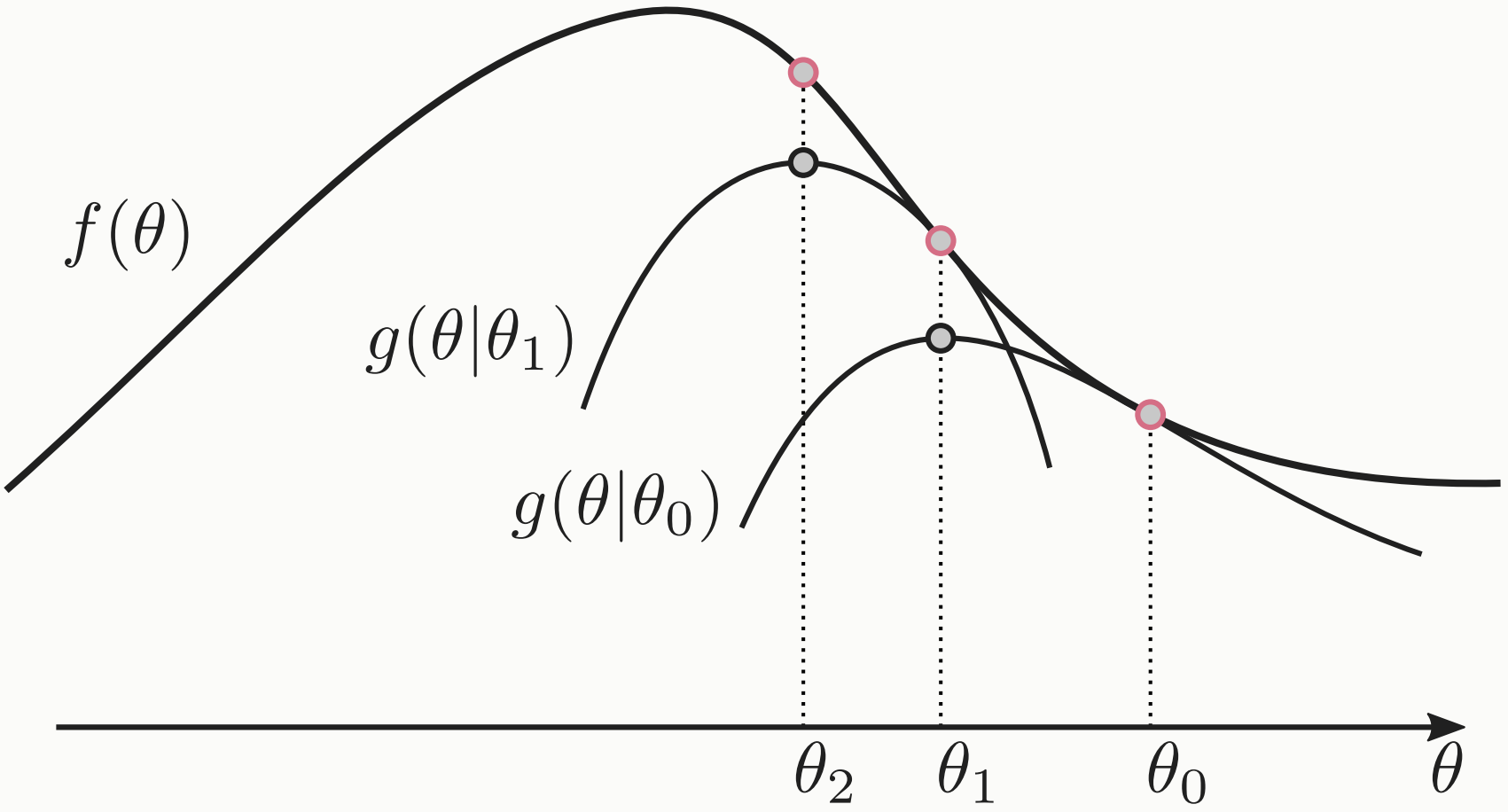
MM アルゴリズムの概要図

MM アルゴリズム



MM アルゴリズムの概要図

MM アルゴリズム



MM アルゴリズムの概要図

minorizer の設計

結果として，平均対数尤度 $f(\boldsymbol{L})$ の minorizer として次を得る：

行列式点過程のための minorizer

$$g(\boldsymbol{L}|\boldsymbol{L}^{(t)}) = -\frac{1}{M}\sum_{m=1}^{M} \mathrm{tr}\{\boldsymbol{L}^{(t)}\boldsymbol{U}_{\mathcal{A}_m}^{\top}[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}\boldsymbol{L}^{(t)}\boldsymbol{L}^{-1}\} - \mathrm{tr}\{(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\boldsymbol{L}\} + \mathrm{const.}$$

ここで $\boldsymbol{U}_{\mathcal{A}_m}$ は N 次単位行列から $\mathcal{A}_m$ の要素に対応する行のみを残した $|\mathcal{A}_m| \times N$ バイナリ行列.

- $[\boldsymbol{L}]_{\mathcal{A}} = \boldsymbol{U}_{\mathcal{A}}\boldsymbol{L}\boldsymbol{U}_{\mathcal{A}}^{\top}$ と書ける

👍 $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ は凹関数

minorizer の最大化

minorizer $g(\boldsymbol{L}|\boldsymbol{L}^{(t)})$ の最大化は，一次の最適性条件から

$$-\boldsymbol{L}(\boldsymbol{L}^{(t)}+\boldsymbol{I})^{-1}\boldsymbol{L}+\boldsymbol{Q}_M^{(t)}=\boldsymbol{O}$$

$$\boldsymbol{Q}_M^{(t)}=\boldsymbol{L}^{(t)}\left(\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{U}_{\mathcal{A}_m}^{\top}[\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1}\boldsymbol{U}_{\mathcal{A}_m}\right)\boldsymbol{L}^{(t)}$$

を満たす $\boldsymbol{L}$ を求めるという問題に帰着

これは CARE[1] という二次の行列方程式の特殊形

- 👍 Schur method などにより $\mathcal{O}(N^3)$ で数値的に解ける

提案法ではこの CARE を繰り返し解いて $\boldsymbol{L}$ の最尤推定値を得る

[1] continuous algebraic Riccati equation.

既存法との関連

既存法と MM アルゴリズム

Mariet and Sra (2015) は不動点アルゴリズムとして更新則

$$\boldsymbol{L}^{(t+1)} = \boldsymbol{L}^{(t)} + \boldsymbol{L}^{(t)} \left(\frac{1}{M} \sum_{m}^{M} \boldsymbol{U}_{\mathcal{A}_m}^{\top} [\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1} \boldsymbol{U}_{\mathcal{A}_m} - (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} \right) \boldsymbol{L}^{(t)}$$

を導いているが，実はこれは minorizer を

$$\begin{aligned} h(\boldsymbol{L} | \boldsymbol{L}^{(t)}) = -\frac{1}{M} \sum_{m=1}^{M} \operatorname{tr}\{ \boldsymbol{L}^{(t)} \boldsymbol{U}_{\mathcal{A}_m}^{\top} [\boldsymbol{L}^{(t)}]_{\mathcal{A}_m}^{-1} \boldsymbol{U}_{\mathcal{A}_m} \boldsymbol{L}^{(t)} \boldsymbol{L}^{-1} \} \\ - \log\det(\boldsymbol{L}) - \operatorname{tr}\{ (\boldsymbol{L}^{(t)} + \boldsymbol{I})^{-1} \boldsymbol{L}^{-1} \boldsymbol{L}^{(t)} \} + \text{const.} \end{aligned}$$

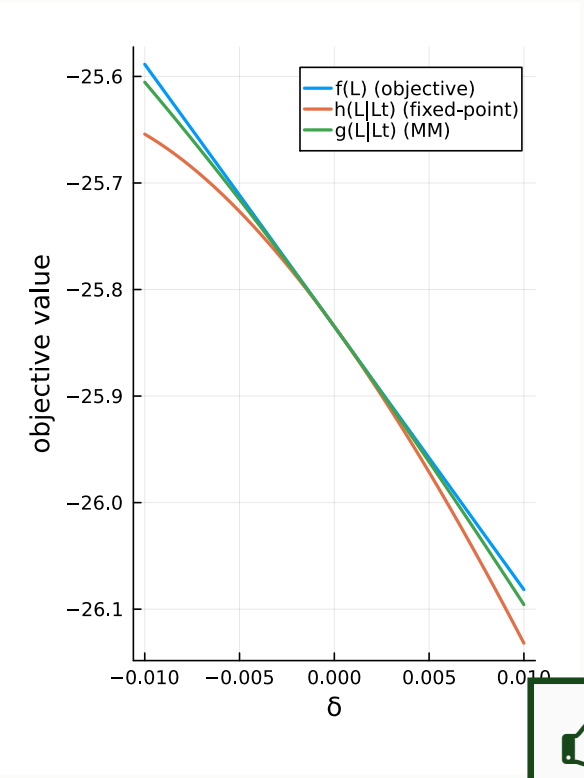
なる非凹関数とおいた MM アルゴリズムとみなせる．

- 👎 minorizer の最適性は必ずしも保証されない

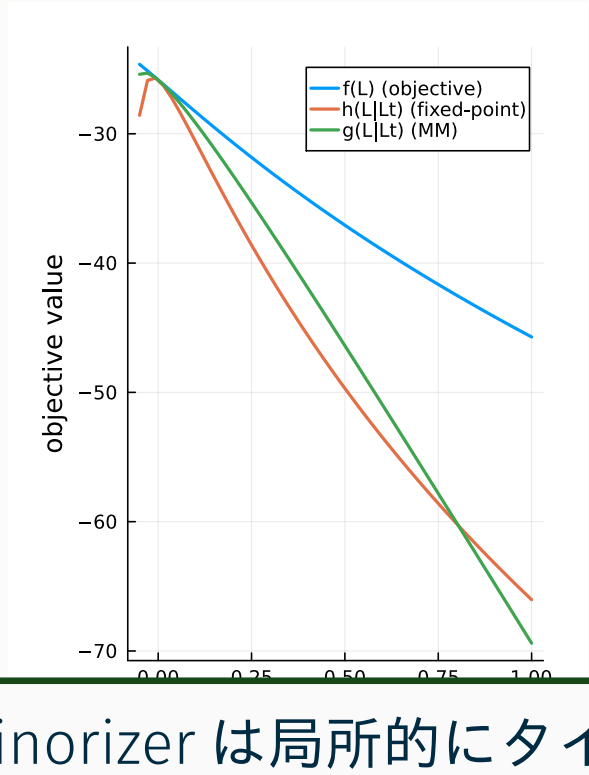
既存法と提案法の関連

提案法と既存法の関連

$\boldsymbol{L}^{(t)}$ 近傍の $\boldsymbol{L}$ に対して，$g(\boldsymbol{L}|\boldsymbol{L}^{(t)}) \geq h(\boldsymbol{L}|\boldsymbol{L}^{(t)})$.



$\boldsymbol{L}^{(t)}$ 近傍



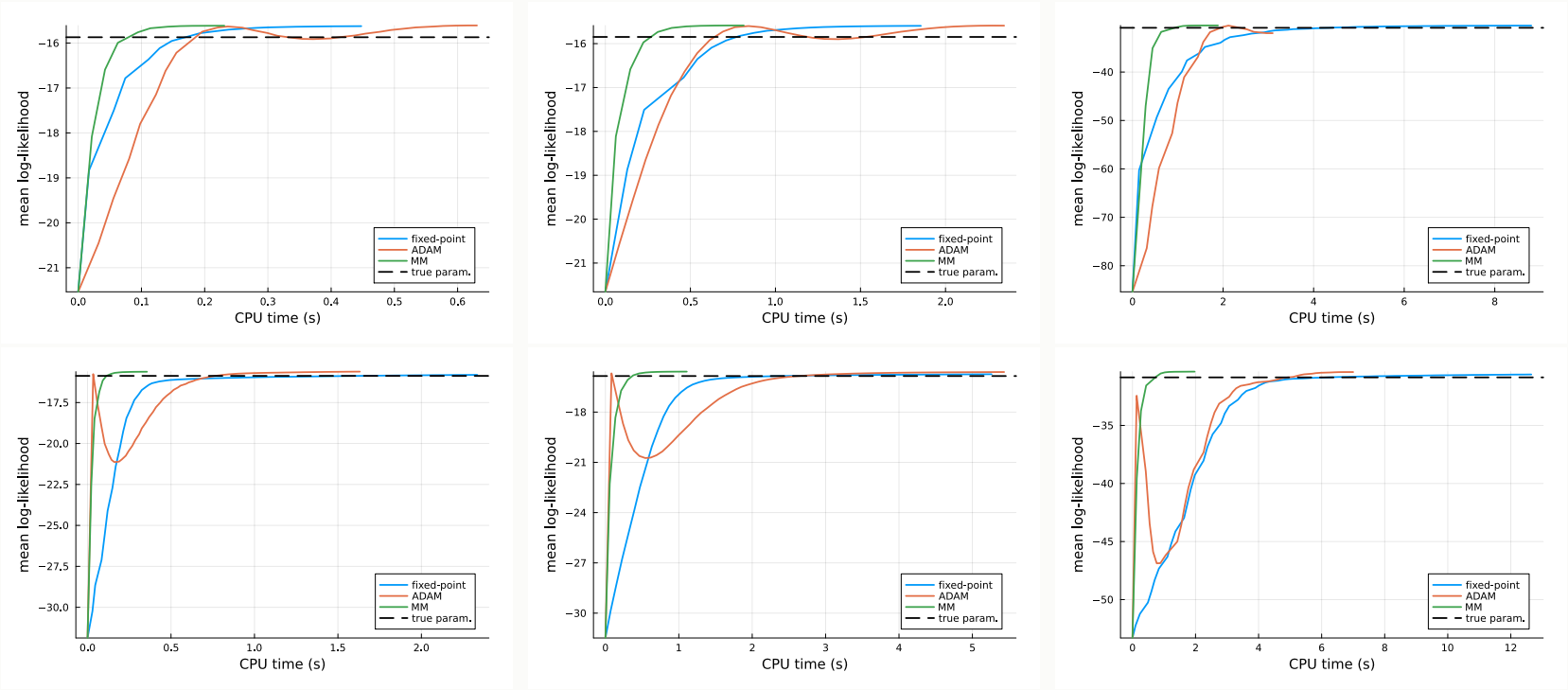
$\boldsymbol{L}^{(t)}$ 非近傍

👍 提案 minorizer は局所的にタイト

実験

実験：toy data

$\boldsymbol{L}^* = \boldsymbol{V}^* \boldsymbol{V}^{*\top}$, $v_{ij}^* \sim \mathcal{U}(0, 10/N)$ から M 個のサンプルを生成

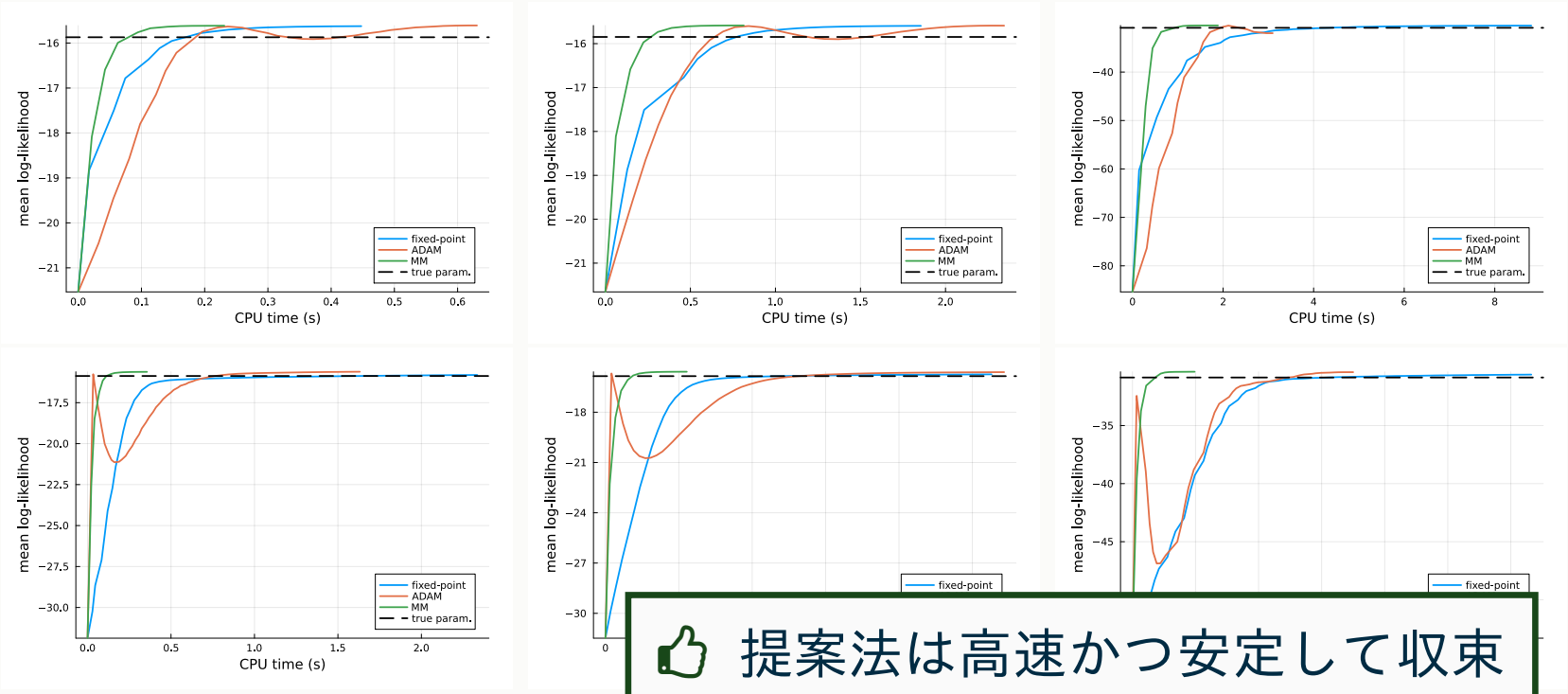


$N = 32, M = 2{,}500$　$N = 32, M = 10{,}000$　$N = 128, M = 2{,}500$

（上段）Wishart 分布から初期化，（下段）一様分布から初期化.
青線：不動点アルゴリズム，橙線：Adam，緑線：提案法.

実験：toy data

$\boldsymbol{L}^* = \boldsymbol{V}^* \boldsymbol{V}^{*\top}$, $v_{ij}^* \sim \mathcal{U}(0, 10/N)$ から M 個のサンプルを生成



👍 提案法は高速かつ安定して収束

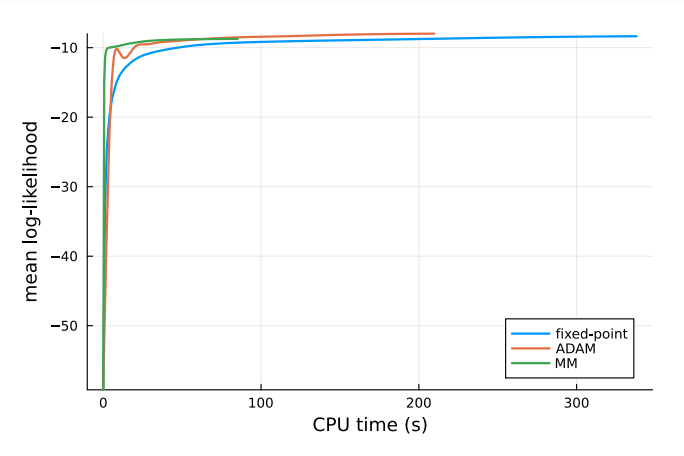
$N = 32, M = 2{,}500$　$N = 32, M = 10{,}000$　$N = 128, M = 2{,}500$

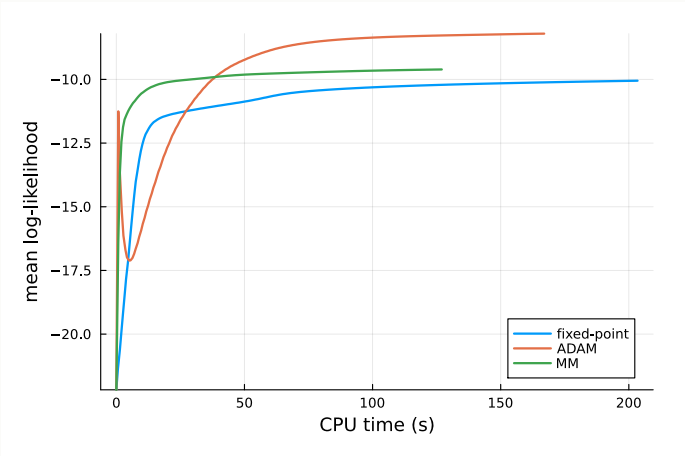
（上段）Wishart 分布から初期化，（下段）一様分布から初期化.
青線：不動点アルゴリズム，橙線：Adam，緑線：提案法.

実験：**Nottingham** データセット

フォーク音楽における各鍵盤の打鍵データ $(N = 88,\ \bar{M} = 6{,}182)$

Method	WISHART Log-likelihood	WISHART Runtime (s)	BASIC Log-likelihood	BASIC Runtime (s)
FP	-8.30 ± 0.22	46.58 ± 3.66	-10.14 ± 0.28	37.26 ± 7.47
Adam	$\mathbf{-7.98} \pm 0.76$	29.36 ± 8.28	$\mathbf{-8.89} \pm 2.96$	29.00 ± 11.91
MM	-9.51 ± 0.25	$\mathbf{24.27} \pm 6.66$	-9.59 ± 0.22	$\mathbf{21.56} \pm 4.24$



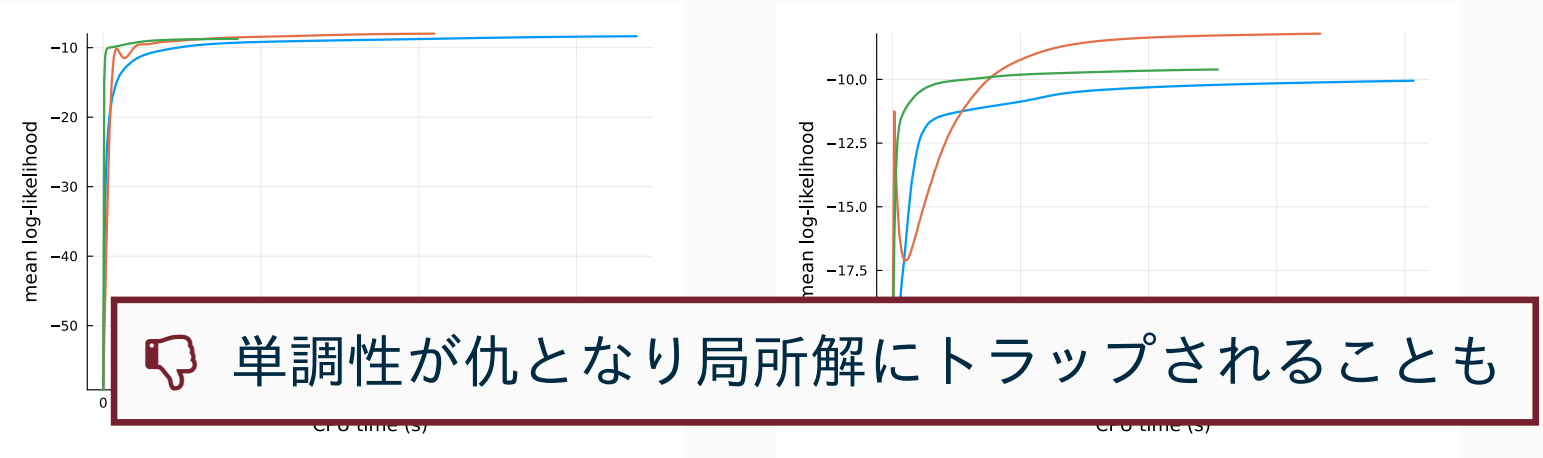


（左）Wishart 分布で初期化，（右）一様分布で初期化．緑線が提案法．

実験：**Nottingham** データセット

フォーク音楽における各鍵盤の打鍵データ $(N = 88,\ \bar{M} = 6{,}182)$

Method	WISHART Log-likelihood	WISHART Runtime (s)	BASIC Log-likelihood	BASIC Runtime (s)
FP	-8.30 ± 0.22	46.58 ± 3.66	-10.14 ± 0.28	37.26 ± 7.47
Adam	$-\mathbf{7.98} \pm 0.76$	29.36 ± 8.28	$-\mathbf{8.89} \pm 2.96$	29.00 ± 11.91
MM	-9.51 ± 0.25	$\mathbf{24.27} \pm 6.66$	-9.59 ± 0.22	$\mathbf{21.56} \pm 4.24$

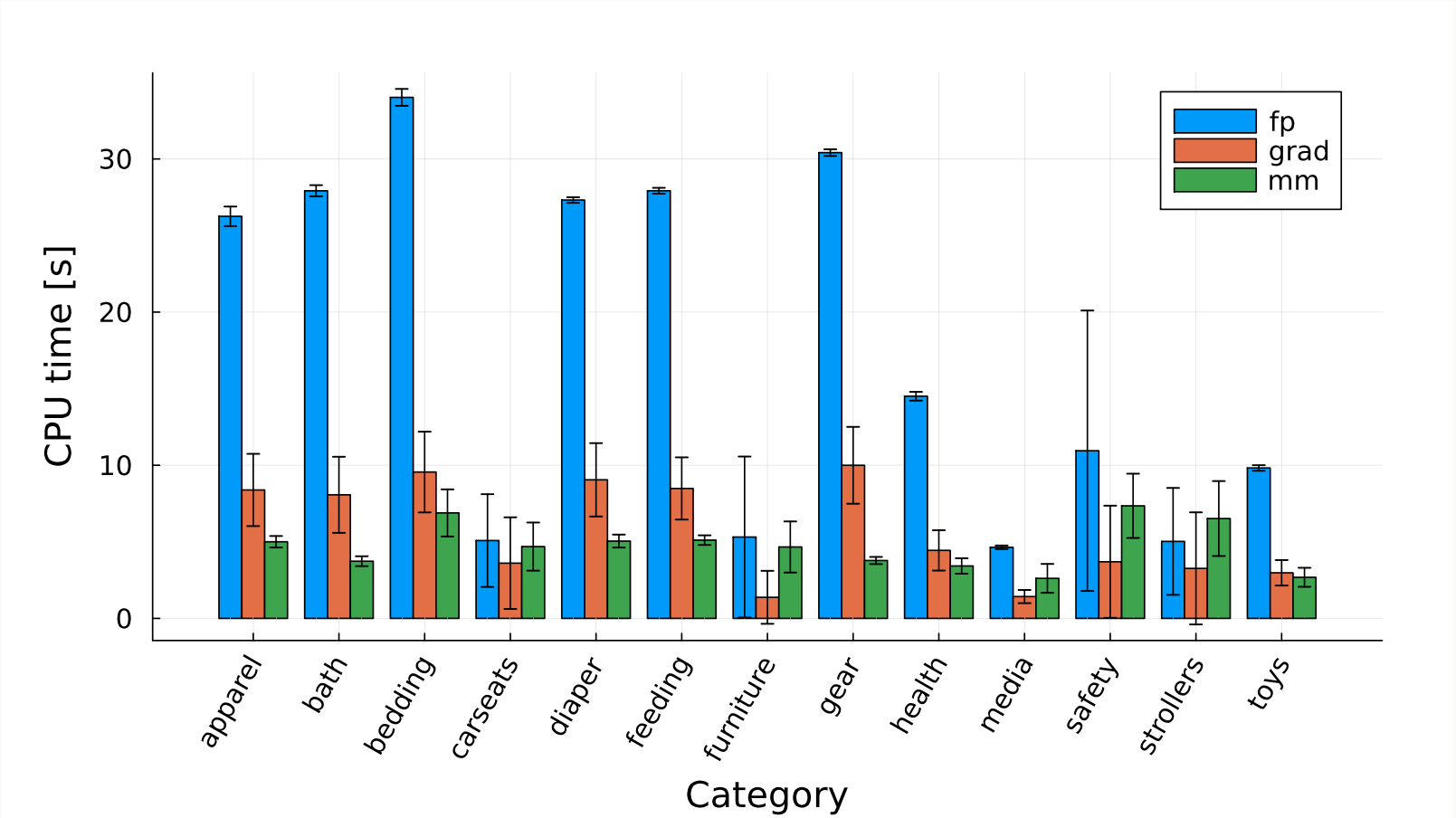


👎 単調性が仇となり局所解にトラップされることも

（左）Wishart 分布で初期化，（右）一様分布で初期化．緑線が提案法．

実験：**Amazon Baby Registry** データセット

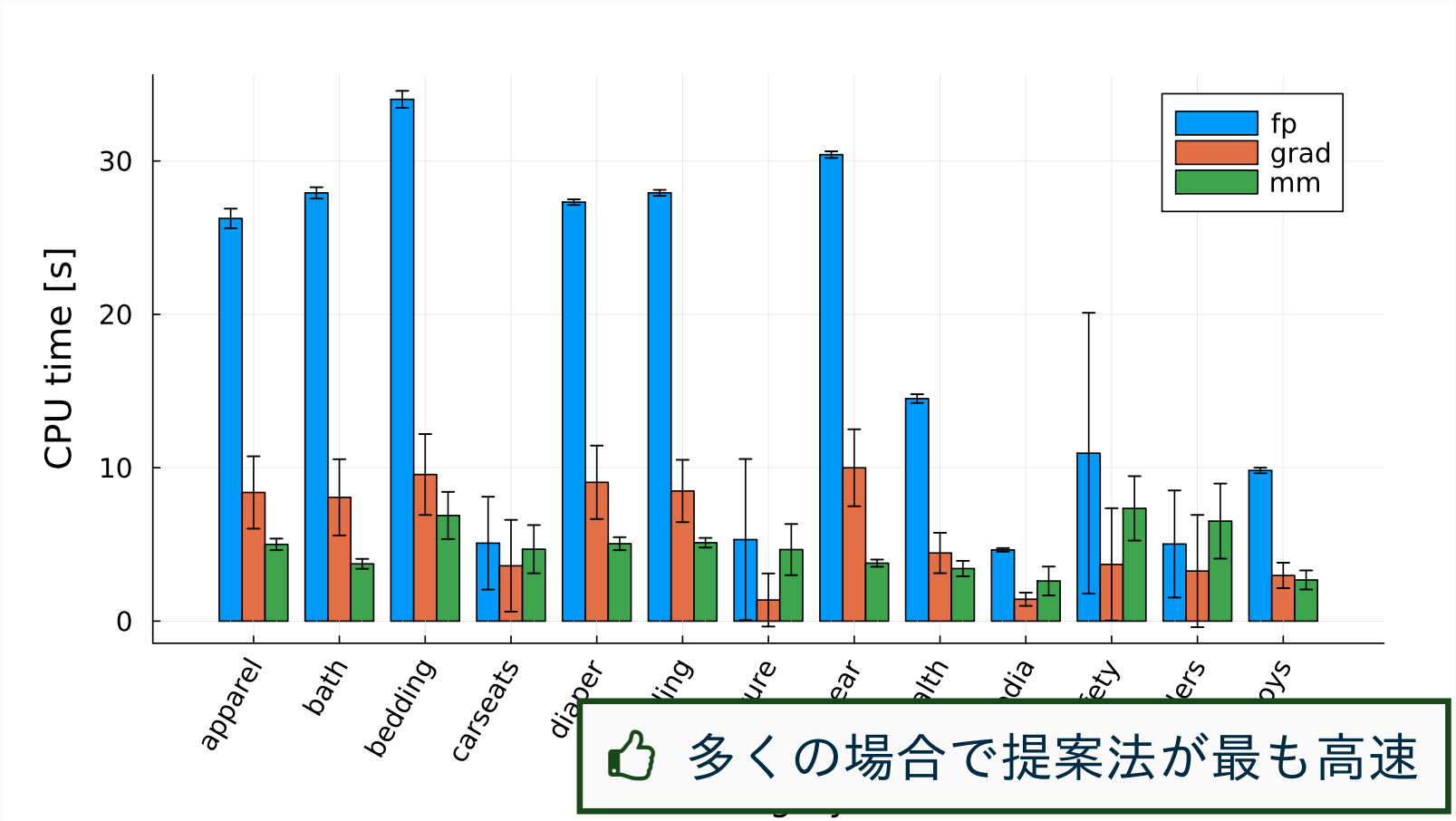
育児関連商品についての欲しいものリスト内の商品を 13 のカテゴリに細分化 $(\bar{N} = 71,\ \bar{M} = 8{,}585)$



各カテゴリに対する計算時間．緑が提案法．

実験：**Amazon Baby Registry** データセット

育児関連商品についての欲しいものリスト内の商品を 13 のカテゴリに細分化 ($\bar{N} = 71,\ \bar{M} = 8{,}585$)



多くの場合で提案法が最も高速

各カテゴリに対する計算時間．緑が提案法．

むすび

むすび

フルランク行列式点過程のための学習アルゴリズムを提案

- 収束速度に優れ，安定で実装も容易[2]かつハイパラフリー
  - MM の性質からアルゴリズムの単調性なども言える
- 局所的に既存法よりタイトな minorizer を与えた
- 最新の結果：実は更新則を一般化して学習を加速できる
  - 単調性は損なわれるが，学習曲線の立ち上がりが改善

[2]CARE のソルバは多くのプログラミング言語で利用できる.

References i

[1] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

[2] Jennifer A Gillenwater, Alex Kulesza, Emily Fox, and Ben Taskar. Expectation-Maximization for Learning Determinantal Point Processes. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[3] Zelda Mariet and Suvrit Sra. Fixed-point algorithms for learning determinantal point processes. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2389–2397, 2015.

[4] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. Low-Rank Factorization of Determinantal Point Processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017.

References ii

[5] Takayuki Osogami, Rudy Raymond, Akshay Goel, Tomoyuki Shirai, and Takanori Maehara. Dynamic Determinantal Point Processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

[6] Zelda E. Mariet and Suvrit Sra. Kronecker Determinantal Point Processes. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[7] Christophe Dupuy and Francis Bach. Learning Determinantal Point Processes in Sublinear Time. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 244–257, 2018.