# 連続最適化問題に対する定数時間アルゴリズム

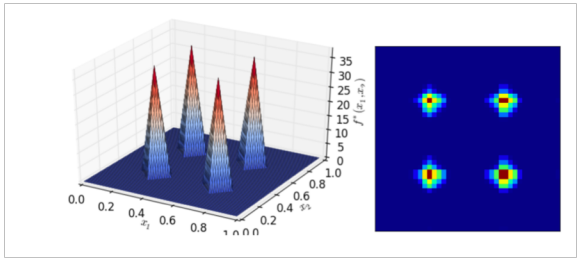
2018/11/6 @ IBIS 2018

吉田悠一
(国立情報学研究所)

# 吉田悠一

所属

- 国立情報学研究所 准教授 ⇒ 理論計算機科学
- ERATO河原林巨大グラフプロジェクト グループリーダー ⇒ グラフアルゴリズム
- Preferred Networks 非常勤シニアリサーチャー ⇒ 深層学習



非連続確率密度推定
with 今泉, 前原 [AISTATS'18]


Spectral Normalization: GAN学習の安定化
with 宮戸, 片岡, 小山 [ICLR'18]

# 理論的考察に基づいた高速アルゴリズム

## 理論研究

定数時間アルゴリズム
（性質検査）

- 入力長に依らない
  計算時間

究極的に高速な
アルゴリズムを
理論的に解明する

## 応用研究

応用的な問題に対する
高速アルゴリズム

- 実世界の入力の
  特徴を活かす
- 理論保証を付ける

数理的な観点から
実用上高速な
アルゴリズムを構築する

# 今日のテーマ

連続最適化問題に対する定数時間アルゴリズム

- 定数時間アルゴリズムとは？

- 二次関数最小化の定数時間近似 (林-吉田, NIPS'16)
- テンソル分解による誤差の近似 (林-吉田, NIPS'17)

- まとめと今後の展開

# 今日のテーマ

連続最適化問題に対する定数時間アルゴリズム

- 定数時間アルゴリズムとは？

- 二次関数最小化の定数時間近似 (林-吉田, NIPS'16)
- テンソル分解による誤差の近似 (林-吉田, NIPS'17)

- まとめと今後の展開

# 性質検査

- 与えられた入力がある性質$\mathcal{P}$を満たすか判定したい
- 入力が巨大な時は線形時間でさえ遅すぎる

性質を「満たす」か「満たすにはほど遠い」かを
区別すれば良いことにして速くならないか？

性質検査
[Blum et al.'93; Rubinfeld and Sudan'96]

多くの性質が入力長に依存しない計算時間 = 定数時間で
判定できる！

# 例：二部グラフ性の検査

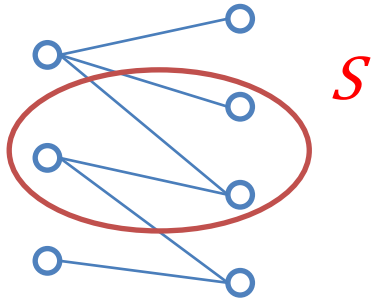
グラフ$G = (V, E)$が二部グラフかどうかを検査したい

<u>アルゴリズム</u>

定数サイズの頂点集合$S \subseteq V$をサンプル

**if** $S$が二部グラフ **then** 受理

**else** 拒否




1. 計算時間は定数
2. $G$が二部グラフの時は必ず受理
3. $G$が二部グラフから遠い時 (e.g., 1%の枝の削除が必要)は高い確率で拒否（非自明）

# 性質検査の研究者としての憂い

理論的に非常に綺麗

- 他の数学との関連 (e.g., 加法的組合せ論)
- 定数時間で検査できる性質の必要十分条件
  - グラフ [AFNS09]
  - 有限体上の関数 [Y. STOC'14] [Y. SODA'16]
  - 制約充足問題 [Y. STOC'11] [Chen-Valeriote-Y. FOCS'16]
  - 関係データベース [Chen-Y. PODS'19]

全く応用されていない

理由（？）

- 離散的な対象ばかり扱っていた
- 判定問題ばかり扱っていた

# 性質検査の研究者としての憂い

連続最適化問題を解こう！

- 応用分野で有用なものが作れるかもしれない
- 理論的に面白い問いが潜んでいるかもしれない

手始めに多項式最小化で定式化できる問題を扱った

# 今日のテーマ

連続最適化問題に対する定数時間アルゴリズム

- 定数時間アルゴリズムとは？

- 二次関数最小化の定数時間近似 (林-吉田, NIPS'16)
- テンソル分解による誤差の近似 (林-吉田, NIPS'17)

- まとめと今後の展開

# 二次関数最小化

二次関数

$$f(\boldsymbol{v}) = \langle \boldsymbol{v}, A\boldsymbol{v} \rangle + \langle \boldsymbol{b}, \boldsymbol{v} \rangle = \sum_{i,j} A_{ij} v_i v_j + \sum_i b_i v_i$$

の”最小値”の近似を定数時間で行う

機械学習・統計・データマイニングに様々な応用

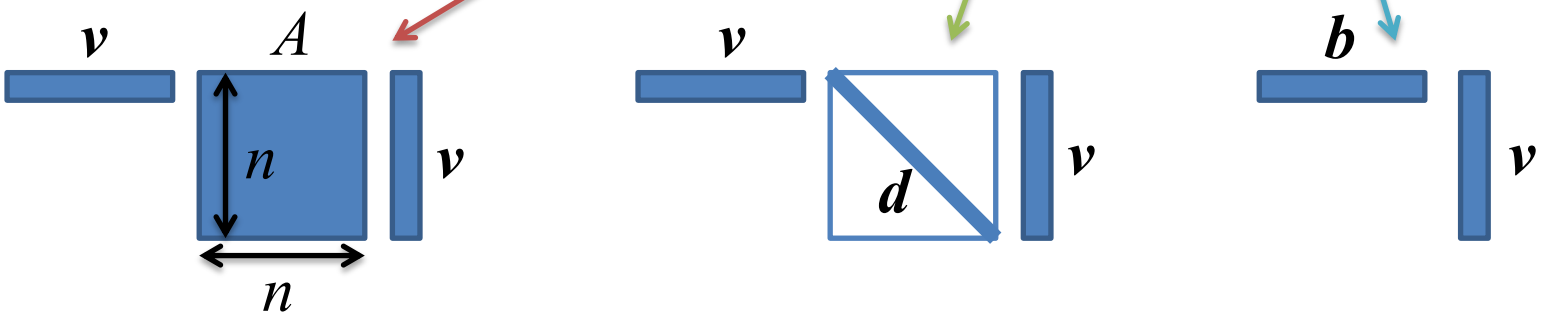
線形回帰, PCA, K-means, …

これらの応用では次元数$n$はしばしば大きい

# 問題設定

目的関数

$$p_{n,A,\boldsymbol{d},\boldsymbol{b}}(\boldsymbol{v}) = \frac{1}{n^2}\left(\langle \boldsymbol{v}, A\boldsymbol{v}\rangle + n\langle \boldsymbol{v}, \mathrm{diag}(\boldsymbol{d})\boldsymbol{v}\rangle + n\langle \boldsymbol{b}, \boldsymbol{v}\rangle\right)$$



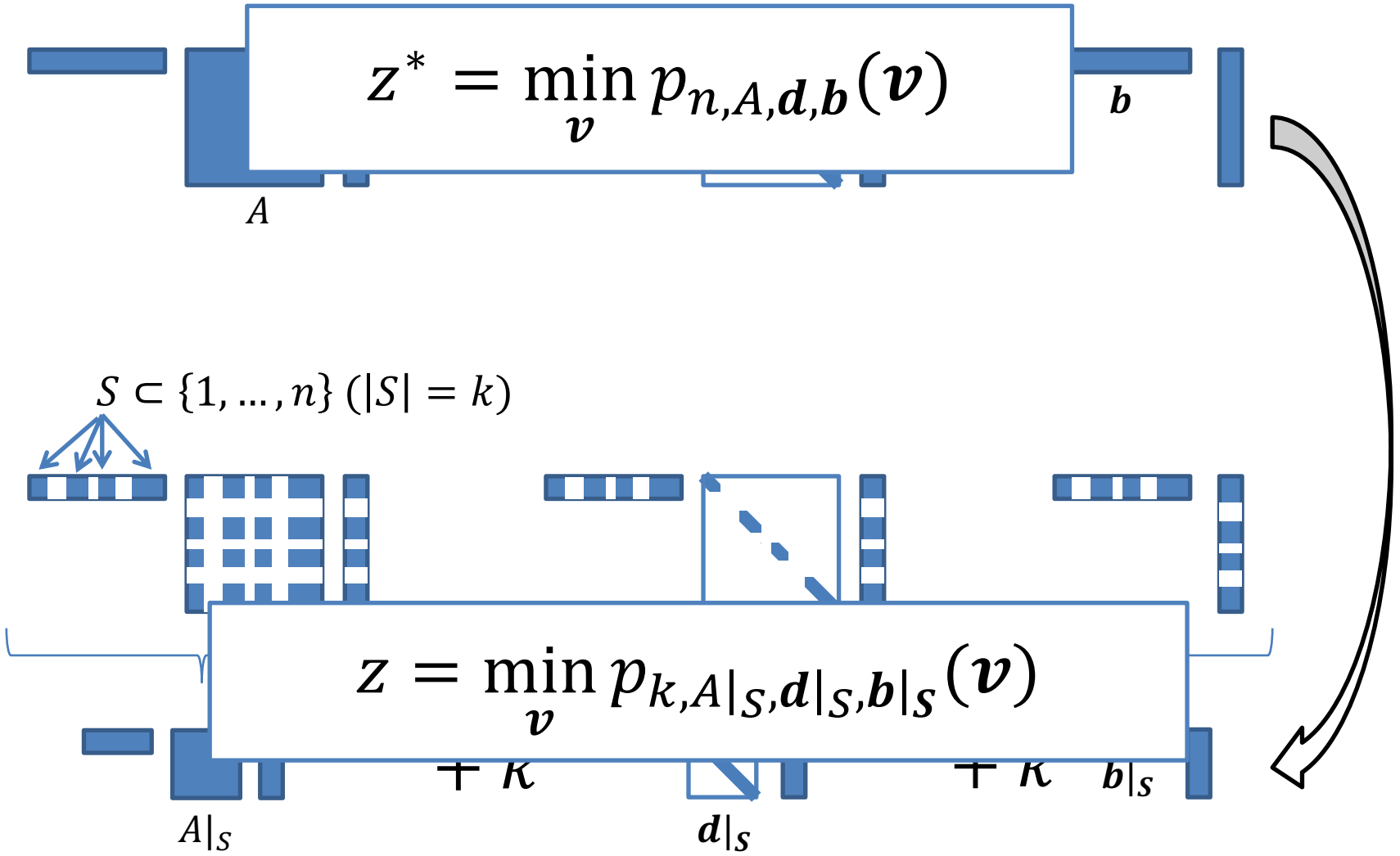

$\|A\|_{\max}\ (:= \max_{i,j}|A_{ij}|) = \|\boldsymbol{d}\|_\infty = \|\boldsymbol{b}\|_\infty = O(1)$ とする

目標

$z^* = \min_{\boldsymbol{v}} p_{n,A,\boldsymbol{d},\boldsymbol{b}}(\boldsymbol{v})$を定数時間で近似する

# 提案アルゴリズム

もとの問題をサンプリングして厳密に解く（簡単！）



# 主定理

[定理] $\boldsymbol{v}^*, \tilde{\boldsymbol{v}}^*$を元の/サンプル後の問題の最適解とする。任意の$\epsilon \in (0,1)$に対し、サンプル数$k = 2^{O(1/\epsilon^2)}$とすると、確率0.99で以下を満たす。

$$|z^* - z| \leq \epsilon \cdot \max\{\|\boldsymbol{v}^*\|_\infty^2, \|\tilde{\boldsymbol{v}}^*\|_\infty^2\}$$

厳密　近似

| 手法 | 計算量 | 注意 |
|---|---|---|
| 厳密計算 | $O(n^3)$ | |
| 確率勾配法 [B04] | $O(nt), t \in \mathbb{N}$ | $t$: 反復回数 |
| Nystrom近似 | $O(nk^2 + k^3)$ | $k$: ランク |
| 乗算型重み更新法 [CHW12] | $O(n/\varepsilon^2)$ | $\boldsymbol{v}$はsimplex上 |
| **提案法** | $\boldsymbol{O\left(k(\epsilon)^3\right) = O(1)}$ | |

# Q & A

サンプル数$k = 2^{O(1/\epsilon^2)}$は大きすぎでは？

⇒ 最近の技術 [JKM18]を使うと多分$\mathrm{poly}(1/\varepsilon)$になる

$L_\infty$ノルムは不自然では？

⇒ 最近の技術 [Levi-吉田 APPROX'18]だと$L_2$ノルムにできる

制約は扱える？

⇒ $\|\boldsymbol{v}\|_\infty \leq 1, \|\boldsymbol{v}\|_2 \leq 1$などに問題無く対応可能

最適解は得られる？

⇒ 解に対するオラクルは作れるが実用的でない

# 証明の雰囲気

[目標] 十分大きな$k$に対して

$$\min_{\boldsymbol{v}\in\mathbb{R}^n} p_{n,A,\boldsymbol{d},\boldsymbol{b}}(\boldsymbol{v}) \approx \min_{\boldsymbol{v}\in\mathbb{R}^k} p_{k,A|_S,\boldsymbol{d}|_S,\boldsymbol{b}|_S}(\boldsymbol{v})$$

元問題とサンプル後の問題が「近い」と言いたい

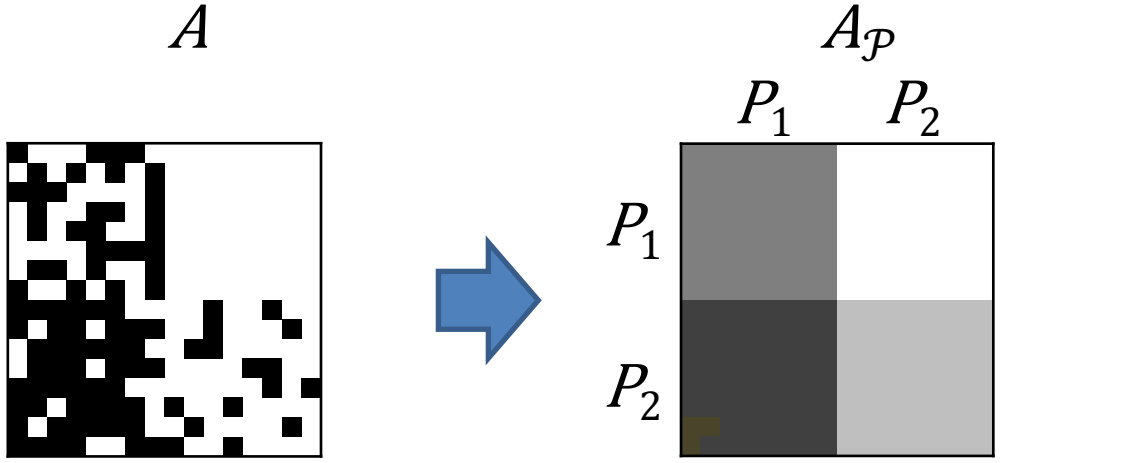
$$A \approx A|_S, \qquad \boldsymbol{d} \approx \boldsymbol{d}|_S, \qquad \boldsymbol{b} \approx \boldsymbol{b}|_S$$

次元が違うので普通のノルムでは距離が測れない

$A$を”次元”の低い行列で近似する

# (弱)正則性補題

[定理] 任意の行列$A \in \mathbb{R}^{n\times n}$は$\{1,2,\dots,n\}$の定数個の部分への等分割$\mathcal{P} = (P_1,\dots,P_k)$を持ち、以下を満たす。



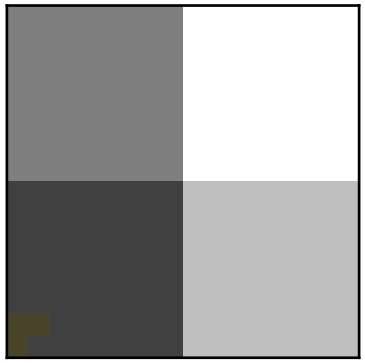

ブロック内を平均化

カットノルム$\|A - A_{\mathcal{P}}\|_{\square}$

$= \sup_{S,T\subseteq\{1,\dots,n\}} |\sum_{u\in S}\sum_{v\in T}(A - A_{\mathcal{P}})_{uv}|$が小さい

# 証明の雰囲気

与えられた行列$A$は以下の様な形をしていると思って良い



定数個の部分しかないので、定数個のサンプルで十分な量の情報が得られる

正確な議論には”Graphon” [L12]を使う

# 実験：ピアソン偽距離の近似

設定：平均の異なる二つのガウシアン分布間のピアソン偽距離をカーネル近似

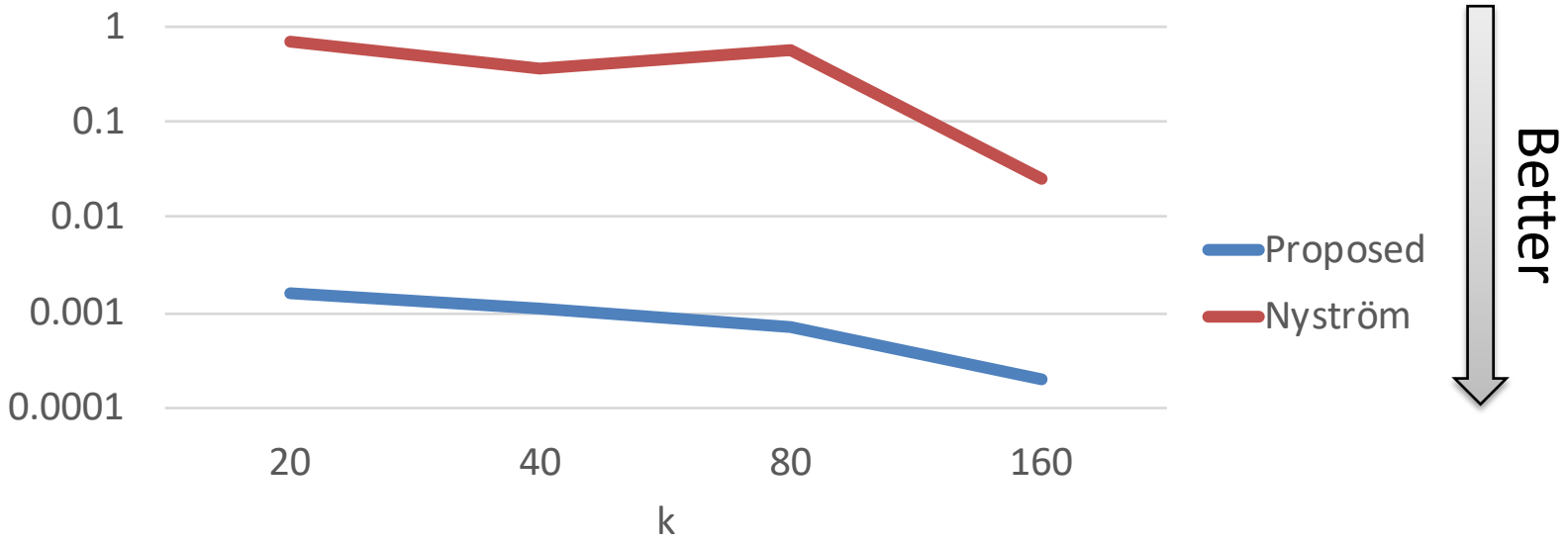
以下の問題に帰着される

$$\min_{v \in \mathbb{R}^n} \frac{1}{2}\langle \boldsymbol{v}, H\boldsymbol{v}\rangle - \langle \boldsymbol{h}, \boldsymbol{v}\rangle + \frac{\lambda}{2}\|\boldsymbol{v}\|_2^2$$

- $H, \boldsymbol{h}$: カーネル関数から決まる行列とベクトル
- $\lambda$: 正則化パラメータ

$n = 5000$で実験

# 精度（絶対誤差）



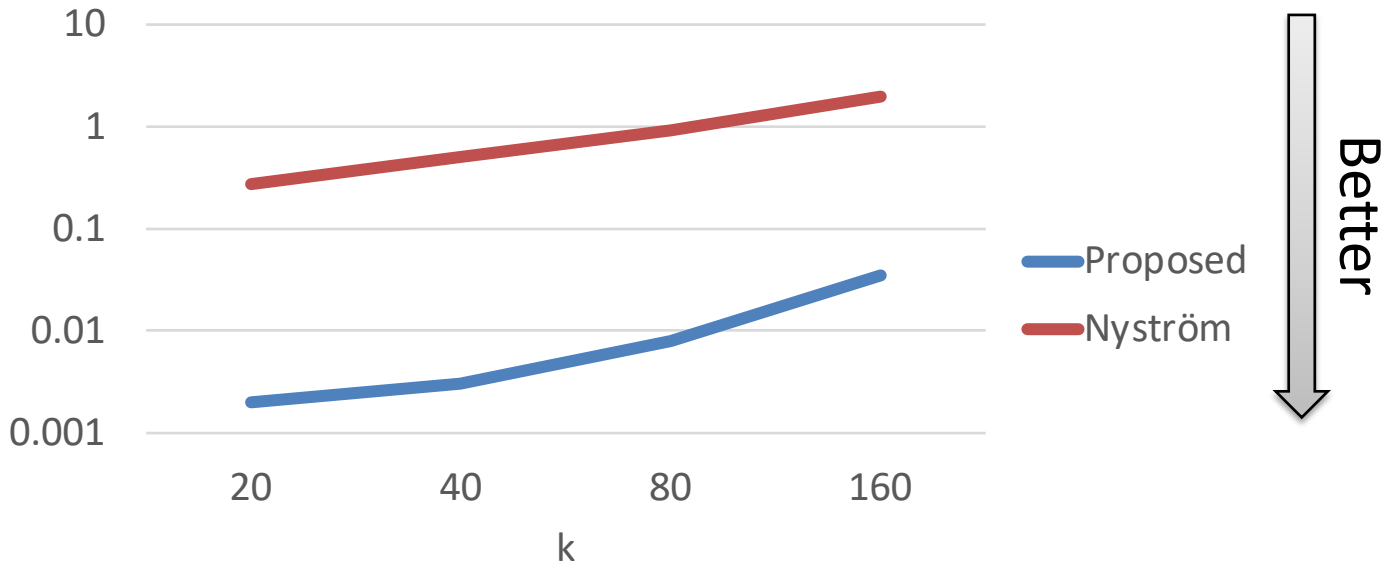

$k$の意味

- 提案手法 = サンプル数
- Nystrom近似 = ランク

同じ$k$だとNystrom近似よりも精度が良い

- Nystrom近似は二次関数最小化の為に設計されていない

# 実行時間




- 同じ$k$だとNystromよりも100～1000倍高速
- 定数時間になってないのはメモリアクセス等のせい？

# 今日のテーマ

連続最適化問題に対する定数時間アルゴリズム

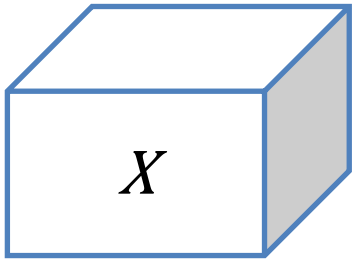
- 定数時間アルゴリズムとは？
- 二次関数最小化の定数時間近似 (林-吉田, NIPS'16)
- テンソル分解による誤差の近似 (林-吉田, NIPS'17)
- まとめと今後の展開

# テンソル

元々は物理学で使われていた概念

ここでは単に多次元配列

- 2階のテンソル (= 行列) $X \in \mathbb{R}^{N \times N'}$
- 3階のテンソル$X \in \mathbb{R}^{N \times N' \times N''}$
- 4階のテンソル$X \in \mathbb{R}^{N \times N' \times N'' \times N'''}$




<u>3階のテンソルの例</u>

- 動画: x軸 × y軸 × 時間 → 画素値
- ソーシャルブックマーク: ユーザ × タグ × Webページ → レーティング

# Tucker分解

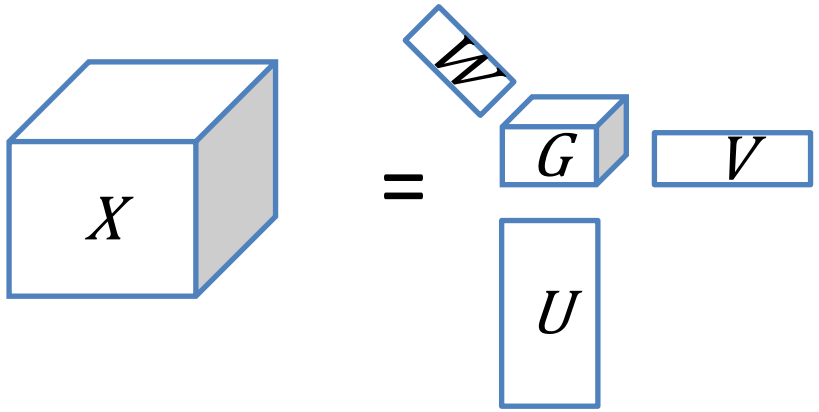
3階のテンソル$X \in \mathbb{R}^{N \times N' \times N''}$が

- コアテンソル$G \in \mathbb{R}^{R \times R' \times R''}$
- 因子行列$U \in \mathbb{R}^{N \times R}$, $V \in \mathbb{R}^{N' \times R'}$, $W \in \mathbb{R}^{N'' \times R''}$

を用いて以下のように分解できたとする

$$X_{ijk} = \sum_{r=1}^{R} \sum_{s=1}^{R'} \sum_{t=1}^{R''} G_{rst} U_{ir} V_{js} W_{kt}$$

⇒ $X$のTuckerランクは$(R, R', R'')$



# Tucker分解の応用

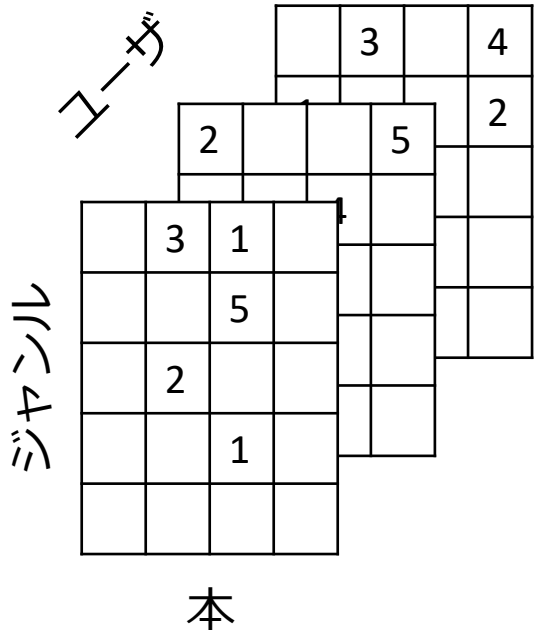
実際のデータが厳密に低ランクなことはない

⇒ ランクを決め打ち、残差が小さければそれで良しとする

$$\min \frac{1}{NN'N''} \sum_{i=1}^{N} \sum_{j=1}^{N'} \sum_{k=1}^{N''} \left| X_{ijk} - \sum_{r=1}^{R} \sum_{s=1}^{R'} \sum_{t=1}^{R''} G_{rst} U_{ir} V_{js} W_{kt} \right|^2$$

Tucker分解の応用

- データの圧縮
- 欠損値の復元 (推薦)



# Tuckerランクの決め方

ランクの決め方

- 交差検証 (欠損値の復元など予測に使う場合)
- 残差 + (AIC or BIC) を最小化

ここでは後者を考える

ランクごとにTucker分解して残差を計算するのは大変

- NP困難 (特異値分解の場合はP)
- 交互最適化等の近似手法も収束まで時間がかかる

残差だけ高速に計算できないか？ → 定数時間アルゴリズム

# Tucker分解に対する定数時間アルゴリズム

$s = s(\varepsilon)$を適切に選ぶ
テンソル$X$の各次元から$s$要素をサンプリング
得られた部分テンソルをTucker分解する
残差を出力

テンソルが小さくなるのでTucker分解が高速になる

正しさの証明は二次関数最小化とほぼ一緒
　ハイパーグラフに対する正則性補題を用いる

# 実験

比較手法

- HOOI (Higher-Order Orthogonal Iteration) [LdMV00]
  $U, V, W$のうち二つを止めて残りの一つについて最適化(特異値分解に帰着される)を繰り返す

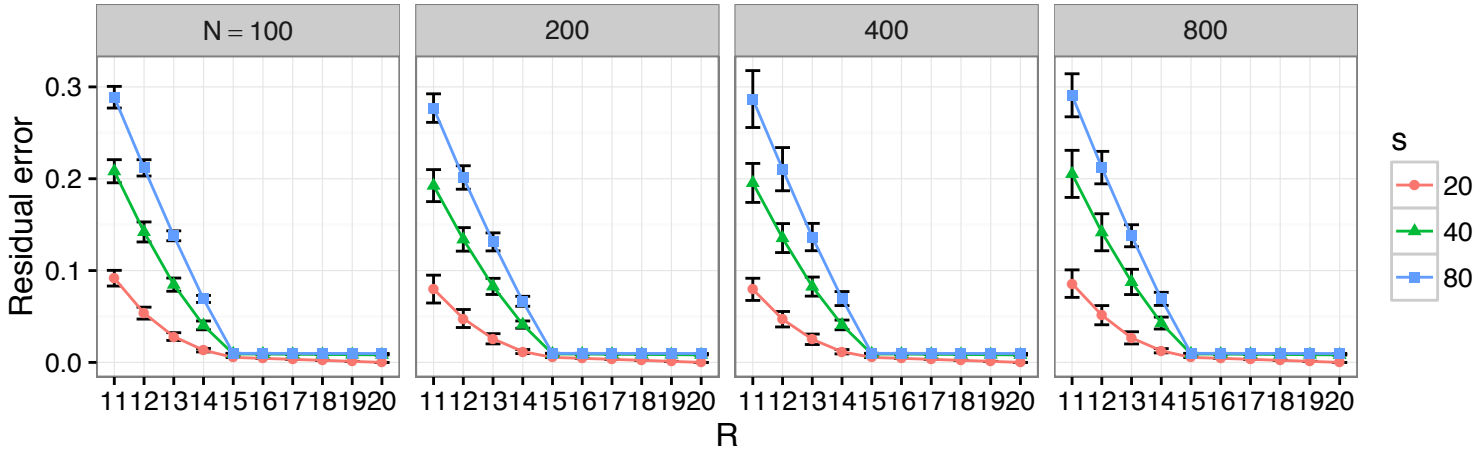
- RandSVD [ZCX14]
  特異値分解を乱択アルゴリズムに置き換えて高速化

- MACH [T10]
  各要素を一定確率で0にし、得られたテンソルを分解

# 人工データにおける精度

ランクを決めて生成したテンソルのランクを復元できるか

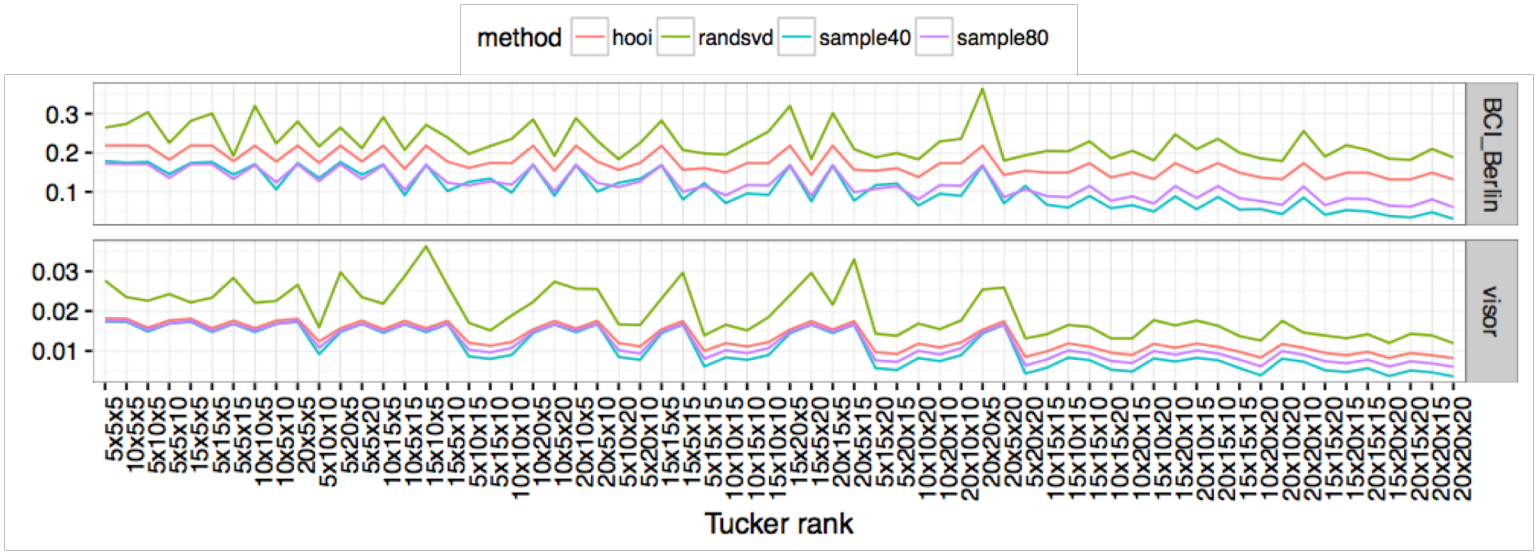
⇒ ランク$(15, 15, 15)$のテンソル$X \in \mathbb{R}^{N \times N \times N}$を生成し、提案手法を適用




どの$N, s$でも正しいランクで残差がほぼ$0$になる

# 実データにおける精度

実データ(主に動画)で残差を正しく近似できるか？




- samples: サンプル数$s$の提案手法
- MACHは誤差が大きすぎる為除外

HOOIを正解とするとRandSVDよりも”それらしい”結果？

# 実データにおける精度

ランクを残差順に並べた時の並びは保存されるか？

HOOIの並びとの相関をKendallの順位相関係数で測定

| | mach0.1 | mach0.3 | randsvd | sample40 | sample80 |
|---|---|---|---|---|---|
| movie_gray | 0.10 | -0.00 | 0.10 | 0.71 | 0.73 |
| EEM | 0.72 | 0.67 | 0.77 | 0.79 | 0.91 |
| fluorescence | 0.04 | 0.09 | 0.28 | 0.61 | 0.77 |
| bonnie | -0.01 | 0.04 | 0.33 | 0.27 | 0.67 |
| fluor | 0.78 | 0.79 | 0.83 | 0.93 | 0.89 |
| wine | -0.07 | -0.01 | -0.02 | 0.04 | 0.15 |
| BCI_Berlin | 0.08 | 0.17 | 0.02 | 0.18 | 0.45 |
| visor | 0.20 | 0.37 | 0.11 | 0.64 | 0.70 |

(1に近いほど相関が高い)

- 提案手法が最も相関が高い
- wineは疎なデータなので精度が悪い

# 計算時間

| | hooi | mach0.1 | mach0.3 | randsvd | sample40 | sample80 |
|---|---|---|---|---|---|---|
| movie_gray | 0.71 | 39.36 | 109.93 | 0.33 | 0.13 | 0.25 |
| EEM | 3447.97 | 16494.45 | 8839.09 | 2212.54 | 0.11 | 0.11 |
| fluorescence | 2.67 | 122.80 | 91.37 | 1.47 | 0.13 | 0.23 |
| bonnie | 9.13 | 102.93 | 72.34 | 2.32 | 0.11 | 0.41 |
| fluor | 3.20 | 47.85 | 149.39 | 1.43 | 0.20 | 0.43 |
| wine | 142.34 | 418.50 | 266.85 | 41.94 | 0.12 | 0.23 |
| BCI_Berlin | 428.13 | 3874.48 | 10258.96 | 82.43 | 0.20 | 0.45 |
| visor | 10034.96 | 27841.67 | 27854.14 | 1950.45 | 0.13 | 0.26 |

- 提案手法が10～1000倍高速
- MACHは疎化するプロセス自体に時間がかかる

# 今日のテーマ

連続最適化問題に対する定数時間アルゴリズム

- 定数時間アルゴリズムとは？

- 二次関数最小化の定数時間近似 (林-吉田, NIPS'16)
- テンソル分解による誤差の近似 (林-吉田, NIPS'17)

- まとめと今後の展開

# まとめ

二次関数最小化・テンソルのTucker分解に対して

- 定数時間で
- 理論保証のある

アルゴリズムを与えた。

証明には正則性補題が大きな役割を果たした。

今後の”自明な”展開：その他の機械学習の問題への応用

- ロジスティック回帰
- カーネル回帰

# 定数時間アルゴリズムと学習理論の“同化”

どちらも巨大な入力の一部から全体を推測しようとする

| | 定数時間アルゴリズム | 学習理論 |
|---|---|---|
| 対象 | グラフ・ブーリアン関数等 | ユークリッド空間上の確率分布 |
| アクセス | クエリ | サンプル |
| 保証 | 所望の誤差を達成するには何クエリ必要？ | 与えられたサンプル数から保証できる誤差？ |

それぞれ独立に発展したので用いる道具が大きく異なる

道具を交換することで新しい知見が得られないか？

# 非連続な確率密度関数

定数時間アルゴリズムの道具 ⇒ 学習理論

[今泉-前原-吉田 AISTATS'18]では、確率密度関数$f$に対して正則性補題を適用することで、$f$が非連続でも収束レートに保証のある確率密度推定ができるようになった。

その他の問題でも理論解析に微分可能性が必要なくなる？
例：カーネル法が使われていた問題 (HSIC等)

課題：高次元の関数を扱うには既存の正則性補題は力不足

# 汎化

学習理論の道具 ⇒ 定数時間アルゴリズム

グラフに対する定数時間アルゴリズムでは、頂点が独立にサンプルされ、その誘導部分グラフが得られる

⇒ 枝に注目すると独立にサンプルされていない

課題：既存の汎化の理論(VC次元, 安定性等)にサンプル非独立性を導入したい

ご清聴ありがとうございました