

武田グループの主な研究成果（2019年度）

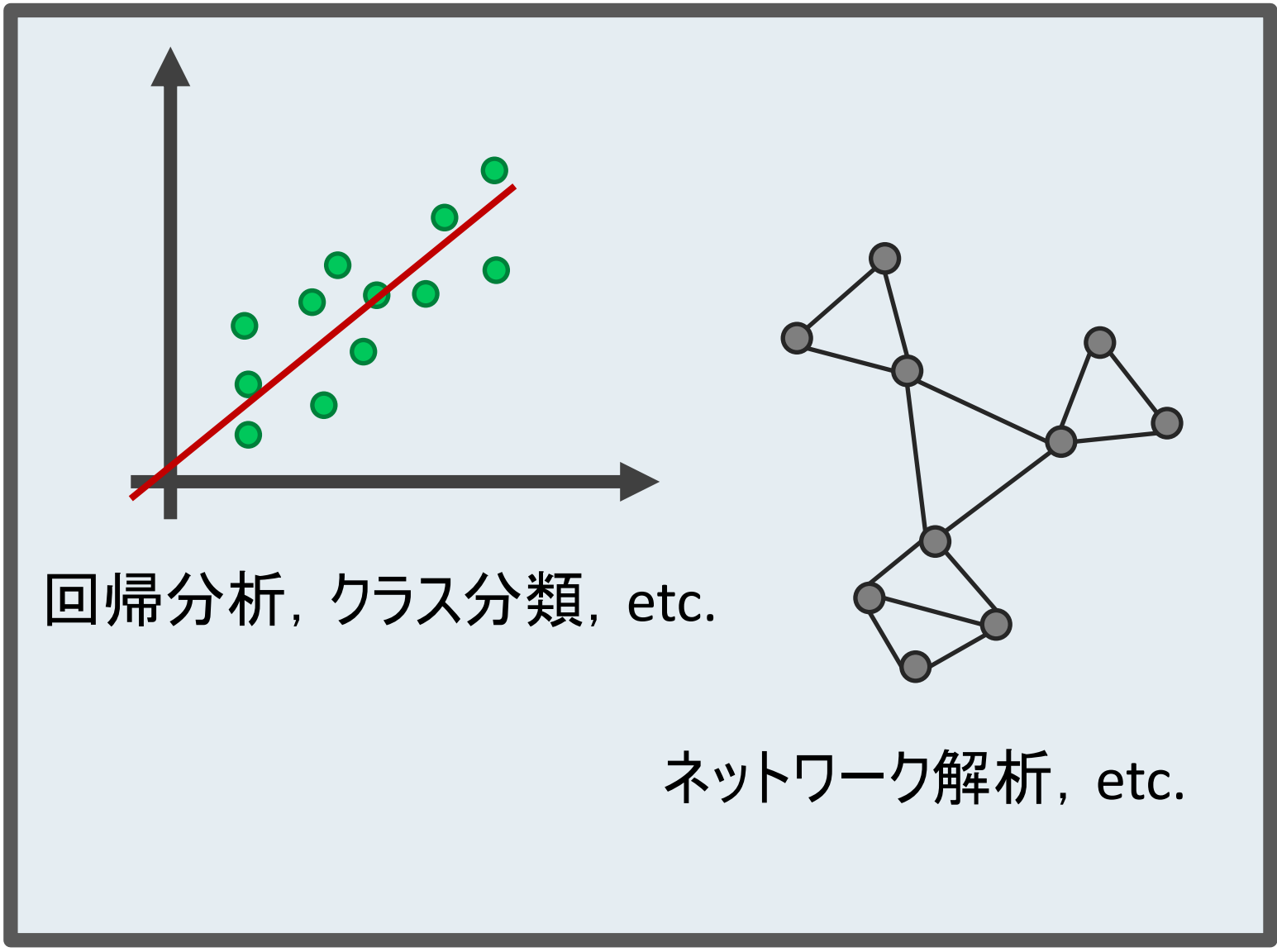
数理最適化（主に**連続最適化**）に関する研究

機械学習やデータマイニングから生じる最適化問題をうまく解き、実社会の問題解決に役立てたい

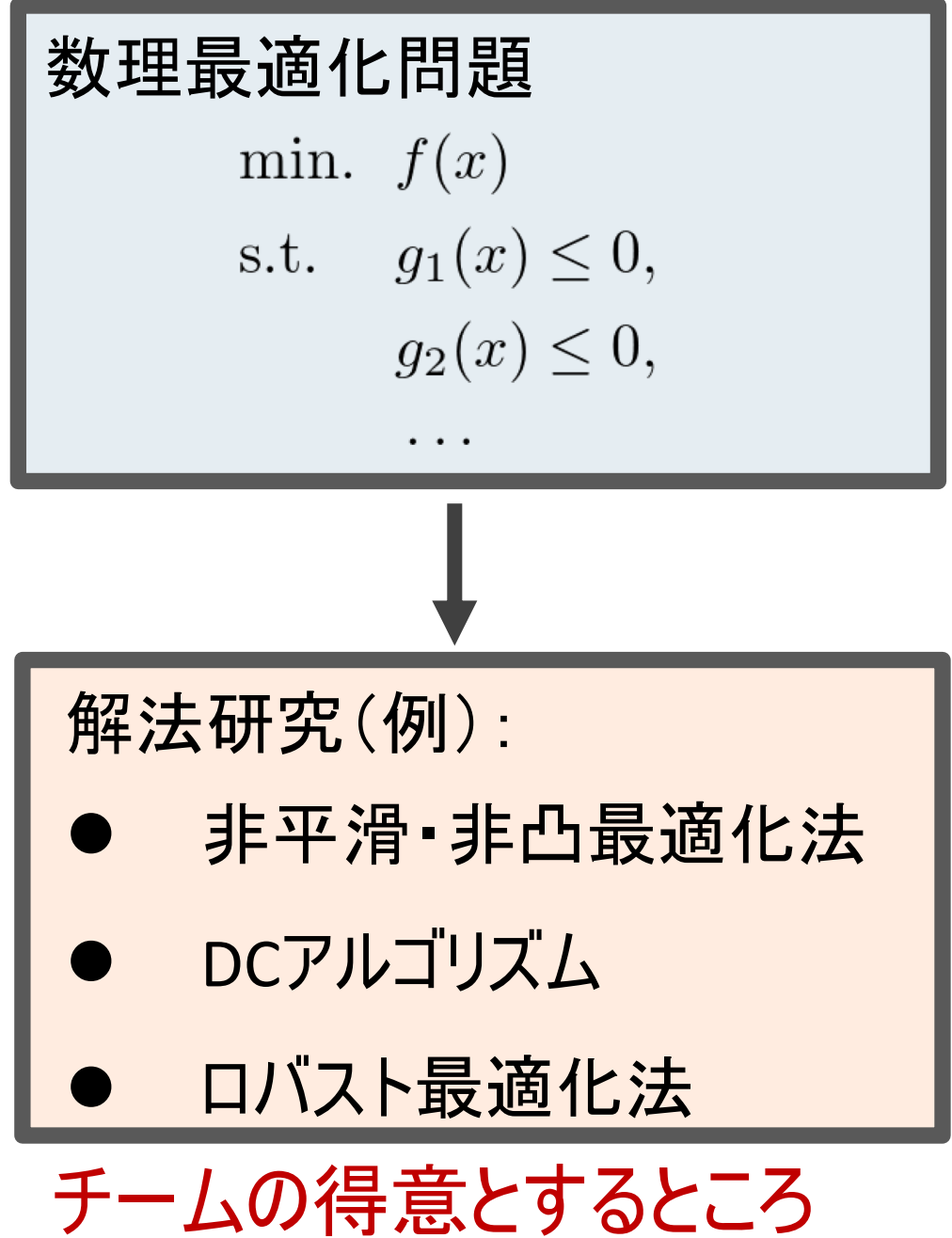
チームの強み：

- 複雑な制約をもつ問題に対する効率的解法
- 非平滑・非凸最適化問題に対する効率的解法
- 理論保証付き DC (Difference-of-Convex) アルゴリズム

機械学習やデータマイニング



モデル化



複数のランク制約をもつ問題に対する DC 近似アルゴリズム

[T. Liu, I. Markovsky, T. K. Pong, and A. Takeda, submitted, 2019]

複数のランク制約をもつ問題：

$$\min_{y=(y_1 \dots y_N)} f(y) \quad \text{s.t.} \quad y \in \Omega_1 \cap \Omega_2.$$
$$\Omega_1 := \{y : \text{rank}(\mathcal{H}_{r_i+1}(y_i)) \leq r_i \quad \forall i\},$$
$$\Omega_2 := \{y : \text{rank}([\mathcal{H}_{r+1}(y_1) \dots \mathcal{H}_{r+1}(y_N)]) = r\}.$$

- $y_i \in \mathbb{R}^n$
- $\mathcal{H}_s(\cdot)$ Hankel 行列
- r_i と r ：正整数
- f ：level-bound かつ勾配は Lipschitz 連続

提案解法 **SDCAM**：

$$\min_{y \in \mathbb{R}^{Nn}} F(y) := f(y) + \delta_{\Omega}(y) + \sum_{i=1}^k \delta_{C_i}(\mathcal{A}_i(y)).$$

ペア (Ω, C_i) は 3 通りの設定が可能

SDCAM は以下の問題を解く： $\min_{y \in \mathbb{R}^{Nn}} F_{\lambda}(y) := f(y) + \delta_{\Omega}(y) + \sum_{i=1}^k \frac{1}{2\lambda} \text{dist}^2(\mathcal{A}_i(y), C_i).$

部分問題を解くには Ω への射影が必要だが、その計算は困難→他の方法は？

↓

$\hat{y}$ の Ω に対する擬似射影 ($u \in \Omega$ に関して)

$$\mathcal{P}_{\Omega}^s(\hat{y}; u) := \{y \in \Omega : \hat{y} - y \in N_{\Omega}(y) \text{ and } \|y - \hat{y}\| \leq \|u - \hat{y}\|\}.$$

既存のソルバー **SLRA** を用いることで得られる理論的な結果も存在

解の**実行可能性**は？

後処理: SDCAM の点から交互擬射影

$$z^{t+1} \in \mathcal{P}_{\Omega_1}^s(x^t; z^t) \text{ and } x^{t+1} \in \mathcal{P}_{\Omega_2}^s(z^{t+1}; x^t) \quad t = 0, 1, \dots$$

理論的成果：

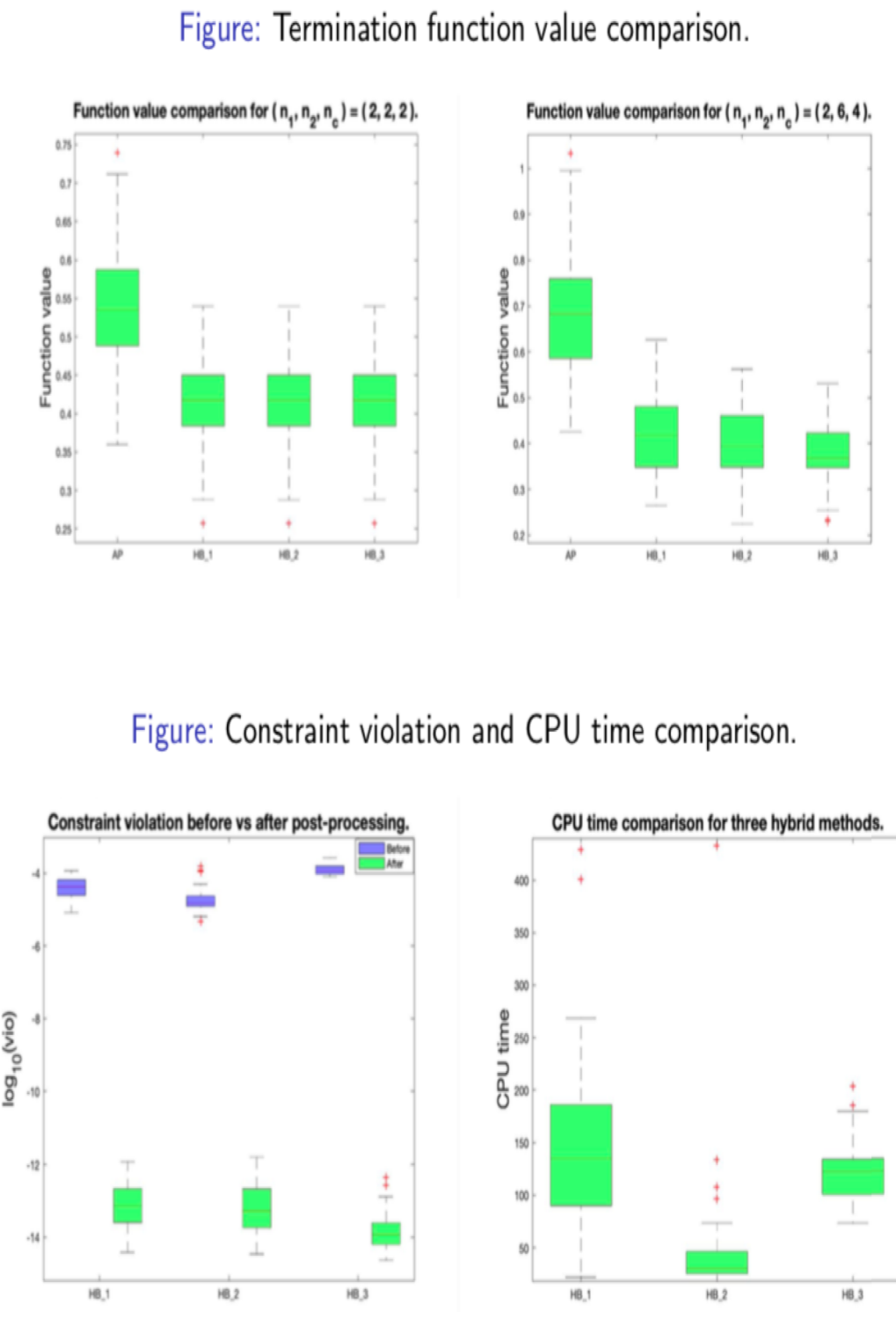
SDCAM の点列は以下を満たす：

$$\text{dist}(\mathcal{A}_i(y^k), C_i) \leq \sqrt{2\lambda_{k-1} f(y^{\text{feas}})}$$

後処理の点列は、以下の条件を満たす

$\bar{y} \in \Omega_1 \cap \Omega_2$ が存在すれば、局所 **R-線形収束**

$$\text{rank}([\mathcal{H}_{r+1}(\bar{y}_1) \dots \mathcal{H}_{r+1}(\bar{y}_N)]) = r, \quad N_{\Omega_1}(\bar{y}) \cap N_{\Omega_2}(\bar{y}) = \{0\}.$$



非平滑・非凸最適化に対する確率的手法

[M. Metel, A. Takeda, ICML, 2019]

[M. Metel, A. Takeda, arXiv:1905.10188, 2019]

問題設定：

$$\min_{w \in \mathbb{R}^d} \Phi(w) := f(w) + g(w) + h(w)$$

where $f(w) := \mathbb{E}_{\xi}[F(w, \xi)]$

- $f(w)$ ：損失関数 (勾配は Lipschitz 連続)
- $g(w)$ ：スパース正則化項 (Lipschitz 連続)
- $h(w)$ ：凸制約 (閉真凸関数)

非負スパース PCA: RCV1 dataset
次元: 804,414 x 47,236

Legend: MBSPA (solid blue line), SDCAM (dotted black line)

劣微分の意味での収束：

$$\mathcal{G}_{\gamma}^{\Phi}(w) := \left\{ \frac{1}{\gamma} (w - \text{prox}_{\gamma h}(w - \gamma s)) : s \in \nabla f(w) + \partial g(w) \right\}$$

収束基準: 期待値評価による ϵ -accurate な解

$$\mathbb{E} [\text{dist}(0, \mathcal{G}_{\gamma}^{\Phi}(w))] \leq \epsilon$$
$$\text{dist}(0, \mathcal{G}_{\gamma}^{\Phi}(w)) = 0 \rightarrow 0 \in \nabla f(w) + \partial g(w) + \partial h(w) \text{ が成り立つ}$$

ミニバッチ確率的近接法 (Mini-batch stochastic proximal algorithm; MBSPA):

$f(w) + g(w)$ の平滑近似に対する確率的勾配

$$\nabla A_{\lambda, M}^k(w^k, \xi^k) = \frac{1}{M} \sum_{j=1}^M \nabla F(w^k, \xi_j^k) + \frac{1}{\lambda} (w^k - \zeta^{\lambda}(w^k))$$

where $\zeta^{\lambda}(w^k) \in \text{prox}_{\lambda g}(w^k)$

アルゴリズムでの主な計算: $w^{k+1} = \text{prox}_{\gamma h}(w^k - \gamma \nabla A_{\lambda, M}^k(w^k, \xi^k))$

計算複雑度：

勾配の計算回数: $O(\epsilon^{-5})$ 近接作用素の計算回数: $O(\epsilon^{-3})$