



信号処理と最適化

早稲田大学 表現工学科 講師

矢田部 浩平





信号処理と数理モデル

最適化アルゴリズム

応用例





▶ 信号処理と数理モデル ◀

最適化アルゴリズム

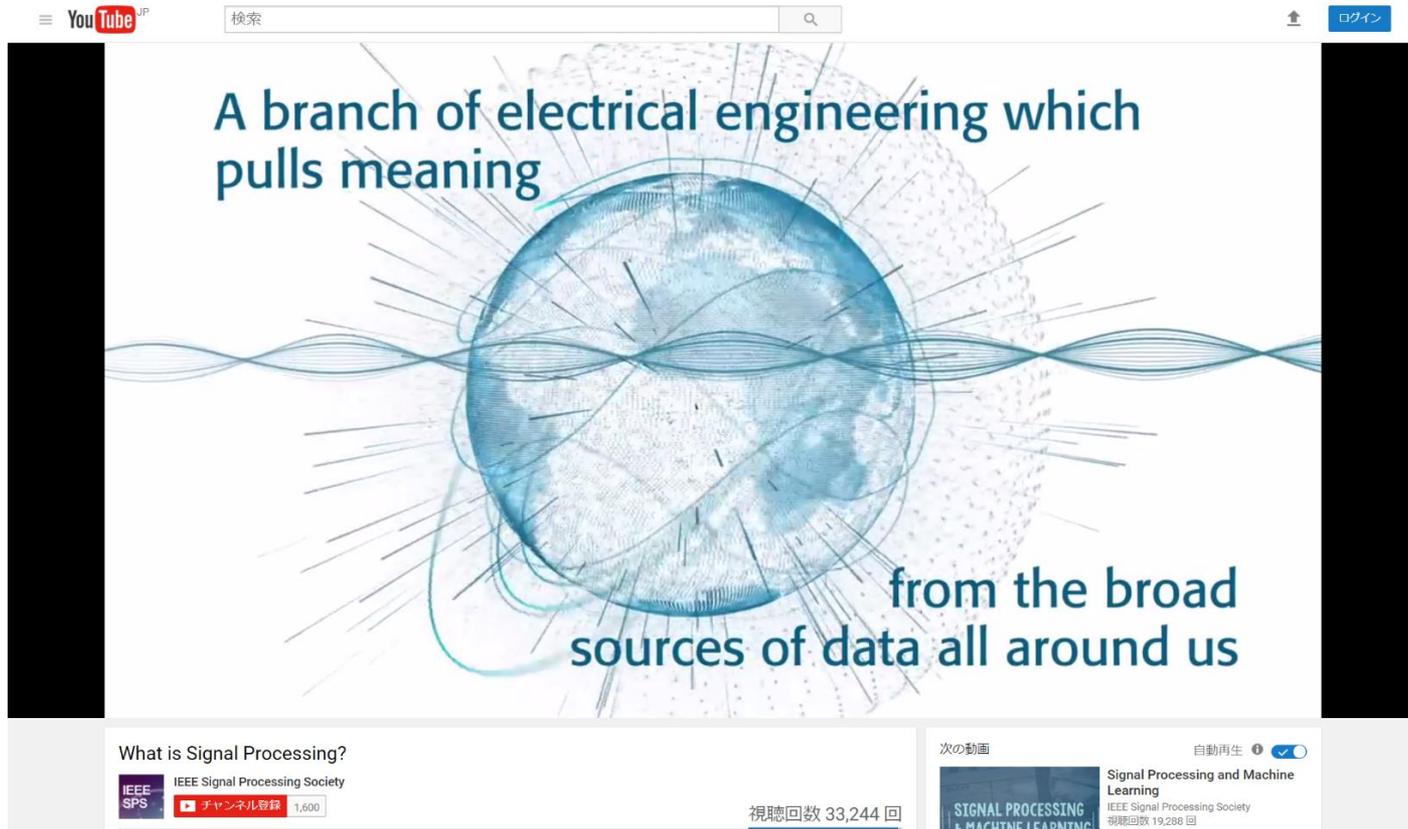
応用例



What is Signal Processing?

IEEE Signal Processing Society (2014.9)

- A branch of electrical engineering which **pulls meaning from the broad sources of data** all around us



The image shows a YouTube video player interface. The video title is "What is Signal Processing?". The channel is "IEEE Signal Processing Society". The video description includes the text: "A branch of electrical engineering which pulls meaning from the broad sources of data all around us". The video has 33,244 views. The player also shows a recommended video titled "Signal Processing and Machine Learning" with 19,288 views.

<https://www.youtube.com/watch?v=R90ciUoxcJU>

信号処理の常套手段



信号処理には2段階のプロセスがある

- **モデル化・定式化**：扱う対象を数学の問題で表現する
- **最適化**：数理モデルで表現された問題を解く

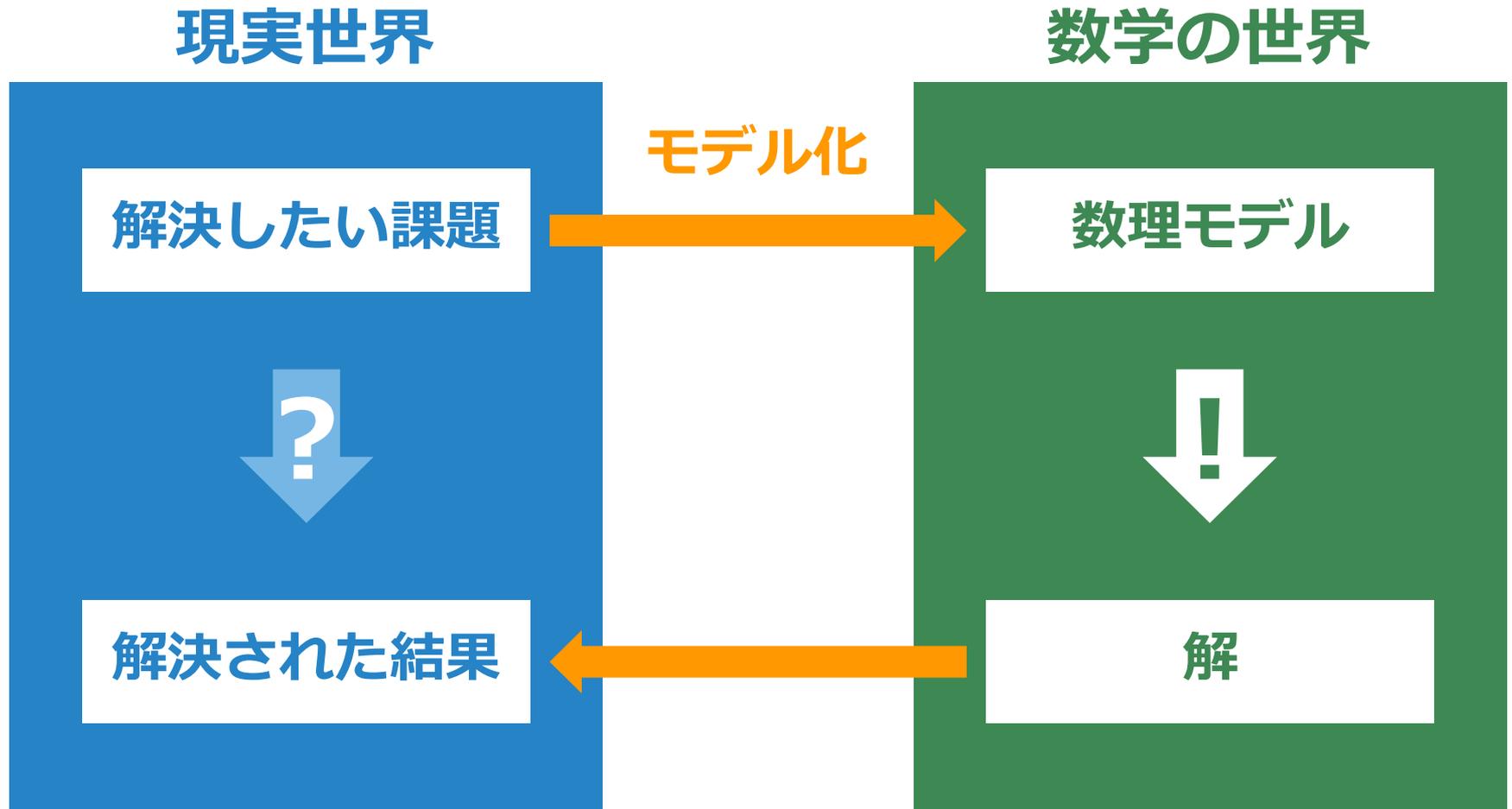
モデル化によって数学を使えるようになる

- 数学は問題を記述し解くための便利なツール
⇒ 数学の言葉になってしまえばあとは解くだけ

最適化は問題を解くための数学的手段

- 数学的に直接解くことが不可能な難しい問題が実用上よく出てくる
⇒ 直接解けない問題を近似的に解くための方法論が最適化

現実世界のモデル化



モデルとは

オックスフォード英語辞典によると

- **A simplified description**, especially a mathematical one, of a system or process, **to assist calculations and predictions.**

Definition of *model* in English:

model 



NOUN

- 3 A simplified description, especially a mathematical one, of a system or process, to assist calculations and predictions.

'a statistical model used for predicting the survival rates of endangered species'

+ More example sentences

+ Synonyms

<https://en.oxforddictionaries.com/definition/model>

なぜモデルを用いるか

世の中の現象は非常に複雑で難しい

- 例：1 m³の室内の音は**十秊個の空気分子の挙動**（千兆の百億倍）
∴ 空気分子の位置を格納するのに10000000000000000TBの記憶容量が必要

現象を簡単な数式で表現したものがモデル

- 数学のツールが使えるように**都合の良い定式化**をする
⇒ 空気分子は離散的な対象だが微分できるように空気を連続関数で扱う
- 考えたいことが表現できていればそれで良い
⇒ 「モデルの正確さ」と「問題の解きやすさ」は一般にトレードオフ
- モデルと現実との誤差を**モデル誤差**と呼ぶ
⇒ **全てのモデルは嘘モデル**

モデルは間違っってて当たり前

様々なモデルが存在



何を根拠にモデル化するか

- 同じ現象を異なる無数のモデルで表現可能
⇒ 信号処理にとって都合の良いモデルとは何か？

例：物理モデル

- 物理的な根拠に基づいた定式化（いかに単純でも元が物理ならOK）
⇒ 無限次元の場合もあれば離散力学系を考える場合もある

例：統計モデル

- データの統計的な根拠に基づいた定式化（複雑な現象を大局的に俯瞰）

例：最適化モデル

- 最適化に都合の良いコストを設計して解集合によりデータを定式化

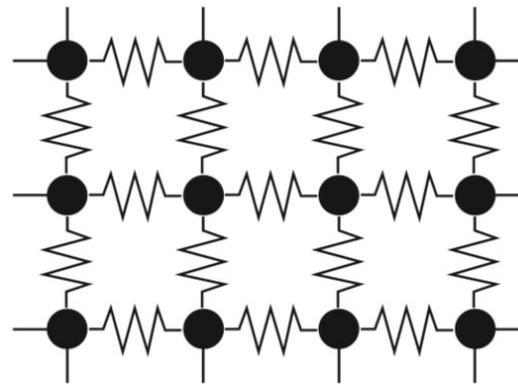
物理モデル

物理的な根拠があれば何でも物理モデル

- 多くの場合は連続モデル
 - ⇒ **波動方程式**やその**境界積分方程式**など（微分方程式・積分方程式）
 - ⇒ 最初から離散的な場合もあり得る
- 大胆な近似を含んでも物理的な根拠に基づけば物理モデル
 - ⇒ バネ・マス系など単純なシステムの集まりで全体を近似する等



弦の近似モデル



膜の近似モデル

統計モデル

Pを知るのとは不可能

確率密度関数 f は仮定

現象を説明

確率現象

モデル

確率変数 X は
未知の法則 P に従う
 $X \sim P$

モデル化
→
(単純化)

未知パラメータ θ に
確率密度関数 f が依存
 $f(x; \theta)$

↓ サンプルング

↑ 推定

観測データ

確率変数 X に対する
無作為標本 \mathbf{x} を観測
 $\mathbf{x} = (x_1, \dots, x_N)$

推定量を
→
計算

推定値 $\hat{\theta}$ が
パラメータ θ を推定
 $\hat{\theta} = g(\mathbf{x})$

観測前は $\mathbf{X} = (X_1, \dots, X_N)$

$g(\mathbf{X})$ は推定値 $\hat{\Theta}$ を与える推定量

モデル化と複雑度

1つの対象は何通りにもモデル化できる

- 1種類の現象に対して**何種類でもモデルは作れる**（全て嘘モデル）
 - ⇒ モデル化によって長所・短所が分かれる（精度 vs 速度とか）
 - ⇒ **どのモデルが最も良いモデルか**の探求が信号処理研究の一側面

あまり複雑なモデルは良くない

- ほとんどのモデルは**パラメータ**によって特徴付けられる
 - ⇒ データからパラメータを推定することで処理を実現
- パラメータが多いモデルは複雑なモデル
 - ⇒ 一般にモデルが**複雑なほどパラメータの推定が難しい**
 - ⇒ 逆にモデルが単純すぎるとモデル誤差が大きくなりがち

モデル化と複雑度

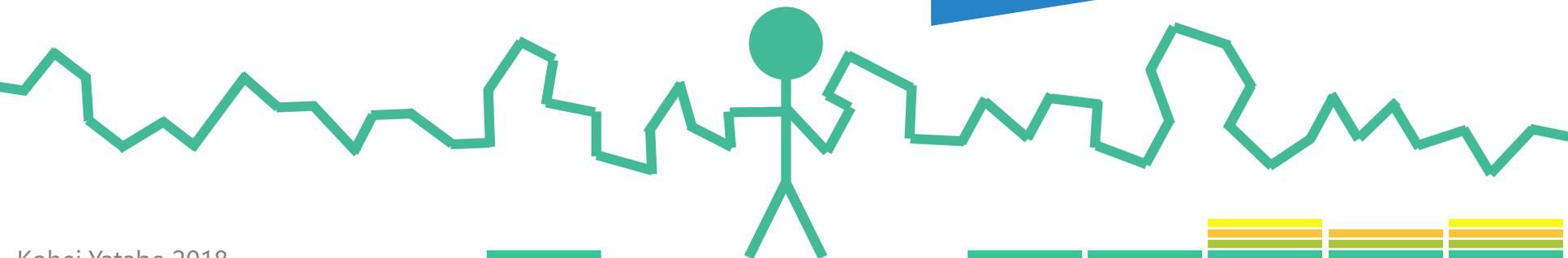
単純すぎるモデルは対象を表現できない

- パラメータが少ないと推定は簡単
 - ⇒ モデル誤差が大きすぎると推定したパラメータが意味をなさない

複雑すぎるモデルは問題を難しくするだけ

- 現実世界のデータはある程度**単純な構造**をしている
 - ⇒ 実際のデータ以上に複雑なモデルはパラメータの取り方に無駄がある
 - ⇒ 無駄なパラメータのせいで推定が困難になったら本末転倒

うわっ...私のモデル, 複雑すぎ...?



モデル化と複雑度

このモデルがキモいのは「人間はこんな風になり得ない」と知ってるから
同様に「音響的にあり得るモデル」を作らないと音響的にキモいかも

複雑すぎるモデルは問題を難しくするだけ

- 現実世界のデータはある程度**単純な構造**をしている
 - ⇒ 実際のデータ以上に複雑なモデルはパラメータの取り方に無駄がある
 - ⇒ 無駄なパラメータのせいで推定が困難になったら本末転倒

うわっ...私のモデル, 複雑すぎ...?



一般のパラメータ推定問題

パラメータの推定問題の多くは最適化問題に帰着

- パラメータの最適性を定める**コスト関数を定義してコストを最小化**
 - ⇒ コスト関数にマイナスをつけて最大化と考えても良い
 - ⇒ コスト関数 $\mathcal{C}_d(\mathbf{x})$ は観測過程に対する仮定で決まる（二乗誤差など）

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{Minimize}} \mathcal{C}_d(\mathbf{x})$$

コストは実数値関数

多くの信号処理では関数の和をコストにする

- コスト関数が2種類の関数の和で表されることが多い
 - ⇒ データの観測過程に関する**データ項**（ノイズのモデル） $\mathcal{D}_d(\mathbf{x})$
 - ⇒ 推定量に対する願望（**先験情報**）を含めた**正則化項** $\mathcal{R}(\mathbf{x})$

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{Minimize}} \mathcal{D}_d(\mathbf{x}) + \lambda \mathcal{R}(\mathbf{x})$$

最大事後確率 (MAP: Maximum A Posteriori)

ベイズ (Bayes) の定理

- 事後確率密度関数はデータを得た上でのパラメータの確率密度関数
⇒ 事後確率密度関数は条件付き密度関数と事前確率密度関数の積に比例

$$f_{X|Y}(x, y) = \frac{f_{Y|X}(y, x) f_X(x)}{\int_{-\infty}^{\infty} f_{Y|X}(y, x) f_X(x) dx} \propto f_{Y|X}(y, x) f_X(x)$$

多くのMAP推定は正則化付き最適化問題に

- コスト関数の和で書かれる正則化付き最適化問題

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{Minimize}} \quad \mathcal{D}_d(\mathbf{x}) + \lambda \mathcal{R}(\mathbf{x})$$

- **事後確率の最大化**を $-\log$ で考えると上記の**最小化問題**に帰着
⇒ 条件付き密度関数 $\exp(-\mathcal{D}_d(\mathbf{x}))$, 事前確率密度関数 $\exp(-\lambda \mathcal{R}(\mathbf{x}))$
⇒ 事後確率は $\exp(-\mathcal{D}_d(\mathbf{x})) \exp(-\lambda \mathcal{R}(\mathbf{x}))$ に比例 (これを最大化)



▶ 信号処理と数理モデル ◀

最適化アルゴリズム

応用例





信号処理と数理モデル

▶ 最適化アルゴリズム ◀

応用例



定式化された問題をどう解くか

定式化が良くても解けなければ無意味

- 問題を解いた結果が「推定されたパラメータ」
⇒ 問題を定式化しても**解かなければ何も得られない**
- 問題の解きやすさ（難易度）は定式化に大きく依存
⇒ 性能低くても難易度が低い簡単な定式化の方が好まれる場合も

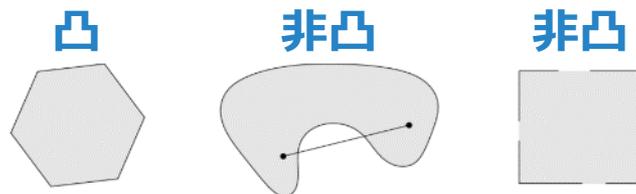
凸最適化問題・非凸最適化問題

- 最適化の意味で解ける問題と解けない問題が存在
⇒ **凸**な問題は（原理的には）解ける，**非凸**な問題は一般に解けない
（性質の良い一部の非凸問題は解ける，固有値問題など）
- 解けない場合は局所解（解ではない）で我慢するしかない
⇒ 性質の良い局所解であれば処理の意味では問題ない

凸集合と凸関数

凸集合とは“凹み”のない集合

- ヒルベルト空間 \mathcal{H} , 部分集合 $\mathcal{C} \subset \mathcal{H}$
- \mathcal{C} が凸であるとは, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{C}, \forall \alpha \in [0, 1]$ に対し以下を満たすこと
⇒ 閉集合であれば**閉凸集合 (closed convex set)** と呼ぶ



$$\alpha \mathbf{x} + (1 - \alpha) \mathbf{y} \in \mathcal{C}$$

凸関数とはエピグラフが凸集合である関数

- 関数 $f : \mathcal{H} \rightarrow \mathbb{R} \cup \{\infty\}$ の**エピグラフ (epigraph)**

$$\text{epi} f = \{(\mathbf{x}, t) \in \mathcal{H} \times \mathbb{R} \mid f(\mathbf{x}) \leq t\}$$

- エピグラフが空でない閉凸集合である関数を**凸関数**と呼ぶ
⇒ 定義域 (effective domain) は $\text{dom} f = \{\mathbf{x} \in \mathcal{H} \mid f(\mathbf{x}) \leq \infty\}$ とする

下半連続な真凸関数の集合

下半連続 (lower semi-continuous)

- 関数 f のレベル集合 (level set)

$$\text{lev}_{\leq \alpha} f = \{\mathbf{x} \in \mathcal{H} \mid f(\mathbf{x}) \leq \alpha\}$$

- レベル集合が $\forall \alpha \in \mathbb{R}$ について閉集合なら**下半連続**であるという
⇒ **下半連続な凸関数の集合**を $\Gamma(\mathcal{H})$ で表す

下半連続な真凸関数の集合 $\Gamma_0(\mathcal{H})$

- 関数 f が $-\infty \notin f(\mathcal{H})$ で $\text{dom} f \neq \emptyset$ なら**真 (proper)** と呼ぶ
⇒ 凸関数が真なら**真凸関数**と呼ぶ
- **下半連続な真凸関数の集合**を $\Gamma_0(\mathcal{H})$ で表す
⇒ $f \in \Gamma_0(\mathcal{H})$ なら $f|_{\text{dom} f}$ は連続

凸最適化問題 (convex optimization problem)

閉凸集合上での凸関数の最小化

- 閉凸集合 \mathcal{C} 上で $f \in \Gamma_0(\mathcal{H})$ を最小化する問題を**凸最適化問題**と呼ぶ

$$\underset{\mathbf{x} \in \mathcal{C} \subset \mathcal{H}}{\text{Minimize}} \quad f(\mathbf{x})$$

- 全ての制約は**指示関数 (indicator function)** で制約なしに書き直せる

$$\underset{\mathbf{x} \in \mathcal{H}}{\text{Minimize}} \quad f(\mathbf{x}) + \iota_{\mathcal{C}}(\mathbf{x}) \quad \iota_{\mathcal{C}}(\mathbf{x}) = \begin{cases} 0 & (\mathbf{x} \in \mathcal{C}) \\ \infty & (\text{otherwise}) \end{cases}$$

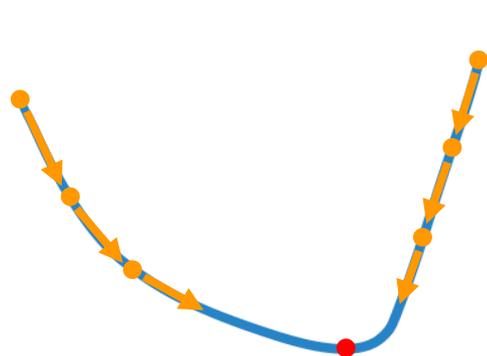
凸最適化問題の大域最適解の存在

- $\lim_{\|\mathbf{x}\| \rightarrow \infty} f(\mathbf{x}) = \infty$ を満たすとき f を**強圧的 (coercive)** と呼ぶ
- $\mathcal{C} \cap \text{dom} f \neq \emptyset$ であって, \mathcal{C} が有界または f が強圧的なら解は存在
⇒ 凸最適化問題なら**全ての局所解は大域最適解**

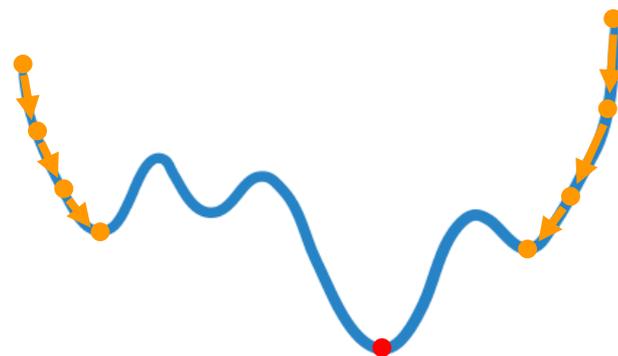
凸最適化問題の解集合

凸なら全ての局所解は大域最適解

- 局所解を求める方法によって大域最適解を得られる
 - ⇒ 関数の局所的な情報を用いて逐次更新するアルゴリズムが利用可能
 - ⇒ どんな初期値からでも大域最適解に到達できる
- 凸最適化問題の**解集合は凸集合**（単元集合とは限らない）
 - ⇒ 複数の解があれば，その間の点も大域最適解（凸集合の定義）



凸関数



非凸関数

勾配降下法 (gradient descent method)

微分可能であれば勾配を下れば良い

- 適当な初期値 $\mathbf{x}^{[0]}$ と適当なステップ幅 $\mu^{[k]} > 0$ で以下を反復

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - \mu^{[k]} \nabla f(\mathbf{x}^{[k]})$$

勾配降下法の収束

- $\forall \mathbf{x}, \mathbf{y} \in \mathcal{H}$ に対して以下を満たす T を **Lipschitz 連続**であると言う
⇒ β を T の **Lipschitz 定数**と呼ぶ

$$\|T(\mathbf{x}) - T(\mathbf{y})\|_2 \leq \beta \|\mathbf{x} - \mathbf{y}\|_2$$

- 勾配 ∇f が β -Lipschitz 連続とする
⇒ 固定ステップ幅 $\mu \in (0, 2/\beta)$ で勾配降下法は解集合に(弱)収束する

2階微分可能な場合

Lipschitz 定数による Hessian の上界

- $f \in \Gamma_0(\mathcal{H})$ の Hessian は単位行列で次の通り上から押さえられる
⇒ $(\beta/2) \|\mathbf{x}\|_2^2 - f(\mathbf{x})$ が凸であることを意味する

$$\nabla^2 f(\mathbf{x}) \preceq \beta I$$

2次関数による上界

- \mathbf{y} まわりの2次の Taylor 展開

$$f(\mathbf{x}) \approx f(\mathbf{y}) + \nabla f(\mathbf{y})^T (\mathbf{x} - \mathbf{y}) + \frac{1}{2} (\mathbf{x} - \mathbf{y})^T \nabla^2 f(\mathbf{y}) (\mathbf{x} - \mathbf{y})$$

- Hessian を βI で置き換えれば2次関数による上界になる

⇒ 上界の最小化解は勾配降下 1 ステップ $\mathbf{x} = \mathbf{y} - (1/\beta) \nabla f(\mathbf{y})$

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T (\mathbf{x} - \mathbf{y}) + \frac{\beta}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \quad \forall \mathbf{x}, \mathbf{y} \in \text{dom } f$$

2階微分可能な場合

Lipschitz 定数による Hessian の上界

- $f \in \Gamma_0(\mathcal{H})$ の Hessian は単位行列で次の通り上から押さえられる
⇒ $(\beta/2) \|\mathbf{x}\|_2^2 - f(\mathbf{x})$ が凸であることを意味する

$$\nabla^2 f(\mathbf{x}) \preceq \beta I$$

2次関数による上界

勾配降下法はステップ幅 $\mu \in (0, 2/\beta)$ で収束するので
 $1/\beta < \mu < 2/\beta$ のときは上界の最小点を通り過ぎている

- Hessian を βI で置き換えれば2次関数による上界になる
⇒ 上界の最小化解は勾配降下1ステップ $\mathbf{x} = \mathbf{y} - (1/\beta)\nabla f(\mathbf{y})$

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T (\mathbf{x} - \mathbf{y}) + \frac{\beta}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \quad \forall \mathbf{x}, \mathbf{y} \in \text{dom } f$$

勾配降下法の弱点

適切なステップ幅を選ばないと発散

- ステップ幅の選択には勾配の Lipschitz 定数が必要
 - ⇒ そもそも勾配が Lipschitz 連続でないと前述のステップ幅は使えない
(各反復毎にラインサーチすることで収束させることは可能)

微分できない点があるとツライ

- 勾配を下るためには全ての $x \in \text{dom} f$ で微分可能でなければいけない
 - ⇒ (特に解集合付近に) 微分できない点があると収束が保証できない
(劣微分可能であれば劣勾配法によって最適化することは可能)

近接点法：ステップ幅の調節が不要な最適化手法

近接作用素 (proximity operator)

- 現在地 \mathbf{y} の近くで関数を最小化する問題

⇒ $f \in \Gamma_0(\mathcal{H})$ なら**最小解は唯一** ($\mathbf{y} \mapsto \text{prox}_{\lambda f}(\mathbf{y})$ は写像として一意)

$$\text{prox}_{\mu f}(\mathbf{y}) = \arg \min_{\mathbf{x}} \left[f(\mathbf{x}) + \frac{1}{2\mu} \|\mathbf{y} - \mathbf{x}\|_2^2 \right]$$

近接点法 (proximal point algorithm)

- 近接作用素を繰り返し適用するアルゴリズム

⇒ $\mu^{[k]} > 0$ が消滅しなければ ($\sum_{k=0}^{\infty} \mu^{[k]} = \infty$ なら) 解集合に(弱)収束

$$\mathbf{x}^{[k+1]} = \text{prox}_{\mu^{[k]} f}(\mathbf{x})$$

- 勾配の Lipschitz 連続性などの仮定が不要, ステップ幅の取り方が自由

近接作用素とは何者か

微分可能であれば勾配降下のようなもの

- 微小なステップ幅を考えれば勾配降下の1ステップと一致
⇒ 2階微分可能で $\nabla^2 f(\mathbf{x}) \succ \mathbf{0}$ のとき $\mu \rightarrow 0$ で

$$\text{prox}_{\mu f}(\mathbf{x}) = \mathbf{x} - \mu \nabla f(\mathbf{x}) + o(\mu)$$

Taylor 展開で近似された関数の勾配降下と一致

- 1次近似された関数 $f_{\mathbf{y}}^{(1)}(\mathbf{x}) = f(\mathbf{y}) + \nabla f(\mathbf{y})^T (\mathbf{x} - \mathbf{y})$
⇒ 勾配は定数 ($\nabla f_{\mathbf{y}}^{(1)}(\mathbf{x}) = \nabla f(\mathbf{y})$) なので明らか Lipschitz 連続

$$\text{prox}_{\mu f_{\mathbf{y}}^{(1)}}(\mathbf{x}) = \mathbf{x} - \mu \nabla f(\mathbf{x})$$

- 勾配降下は「関数を1次近似してから近接作用素を考える」のと同じ
- 2次近似なら Levenberg–Marquardt $\mathbf{x} - (\nabla^2 f(\mathbf{x}) + (1/\mu)I)^{-1} \nabla f(\mathbf{x})$

Moreau–Yoshida正則化をした関数の勾配降下

極小畳み込み (Infimal convolution)

- 関数 $f, g \in \Gamma_0(\mathcal{H})$ の極小畳み込みは以下で定義される凸関数

$$f \square g : \mathcal{H} \rightarrow [-\infty, \infty] : \mathbf{x} \mapsto \inf_{\mathbf{y} \in \mathcal{H}} [f(\mathbf{y}) + g(\mathbf{x} - \mathbf{y})]$$

- 関数 f のMoreauエンベロープ (Moreau–Yoshida正則化) ${}^\mu f$

$${}^\mu f(\mathbf{x}) = (f \square \frac{1}{2\mu} \|\cdot\|_2^2)(\mathbf{x}) = \inf_{\mathbf{y} \in \mathcal{H}} [f(\mathbf{y}) + \frac{1}{2\mu} \|\mathbf{x} - \mathbf{y}\|_2^2]$$

近接作用素は Moreau エンベロープの勾配降下写像

- Moreau エンベロープは (通常 of 畳み込み同様) 関数の近似を与える
⇒ 近似された関数の勾配降下 1 ステップが近接作用素

$$\text{prox}_{{}^\mu f}(\mathbf{x}) = \mathbf{x} - \mu \nabla ({}^\mu f)(\mathbf{x})$$

近接点法の勾配流としての見方

最適化アルゴリズムを微分方程式と考える

- アルゴリズムの各ステップを時間発展だと考える
⇒ 微分可能な $f \in \Gamma_0(\mathcal{H})$ の最小化は以下の**勾配流の時間発展**

$$\frac{d}{dt} \mathbf{x}(t) = -\nabla f(\mathbf{x}(t))$$

勾配流を離散化すると最適化アルゴリズムを得る

- **前進 (forward) Euler 法** $(\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}) / (\Delta t) = -\nabla f(\mathbf{x}^{[k]})$

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - (\Delta t) \nabla f(\mathbf{x}^{[k]})$$

- **後進 (backward) Euler 法** $(\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}) / (\Delta t) = -\nabla f(\mathbf{x}^{[k+1]})$

$$\mathbf{x}^{[k+1]} + (\Delta t) \nabla f(\mathbf{x}^{[k+1]}) = \mathbf{x}^{[k]} \quad \Rightarrow \quad \mathbf{x}^{[k+1]} = (\text{Id} + (\Delta t) \nabla f)^{-1} \mathbf{x}^{[k]}$$

近接点法の勾配流としての見方

最適化アルゴリズムを微分方程式と考える

- アルゴリズムの各ステップを時間発展だと考える
⇒ 微分可能な $f \in \Gamma_0(\mathcal{H})$ の最小化は以下の**勾配流の時間発展**

$$\frac{d}{dt} \mathbf{x}(t) = -\nabla f(\mathbf{x}(t))$$

勾配流を離散化すると最適化アルゴリズムを得る

■ 前進

近接作用素の正体（ ∇f のレゾルベント）

微分法方程式の陰的数値解法に対応してることから

陽的な勾配降下に比べ**安定的な数値積分が可能**

■ 後進

$$\mathbf{x}^{[k+1]} + (\Delta t) \nabla f(\mathbf{x}^{[k]}) = \mathbf{x}^{[k]} \quad \Rightarrow \quad \mathbf{x}^{[k+1]} = \boxed{(\text{Id} + (\Delta t) \nabla f)^{-1}} \mathbf{x}^{[k]}$$

勾配が2項ある場合の勾配流

微分可能な項が2つある場合

- 微分可能な $f, g \in \Gamma_0(\mathcal{H})$ の和 $f + g \in \Gamma_0(\mathcal{H})$ を最小化する勾配流

$$\frac{d}{dt} \mathbf{x}(t) = -\nabla f(\mathbf{x}(t)) - \nabla g(\mathbf{x}(t))$$

前進と後進どちらで離散化すべきか

- 陰的解法の方が一般に数値安定にできるが和のレゾルベントは面倒？
⇒ **作用素の分離 (operator splitting)** による数値スキームを考える
⇒ f に前進 (forward) , g に後進 (backward) を同時に適用

$$(\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}) / (\Delta t) = -\nabla f(\mathbf{x}^{[k]}) - \nabla g(\mathbf{x}^{[k+1]})$$

- 作用素が f と g で**それぞれ分離した更新則**が得られる

$$\mathbf{x}^{[k+1]} = (\text{Id} + (\Delta t)\nabla g)^{-1} (\text{Id} - (\Delta t)\nabla f) \mathbf{x}^{[k]}$$

近接勾配法 (proximal gradient method)

Forward-backward splitting algorithm と

- 近接作用素を勾配降下法に組み合わせて $f(\mathbf{x}) + g(\mathbf{x})$ を最小化
⇒ 勾配ステップを forward step, 近接作用素を backward step と呼ぶ

$$\mathbf{x}^{[k+1]} = \text{prox}_{\mu^{[k]}g} \left(\mathbf{x}^{[k]} - \mu^{[k]} \nabla f(\mathbf{x}^{[k]}) \right)$$

- 勾配 ∇f が β -Lipschitz 連続とする
⇒ 固定ステップ幅 $\mu \in (0, 2/\beta)$ で近接勾配法は解集合に(弱)収束する

特別な場合として射影勾配法や交互射影を含む

- $g = \iota_C$ とすると射影勾配法に

⇒ 指示関数の prox は距離射影

$$\mathbf{x}^{[k+1]} = P_C \left(\mathbf{x}^{[k]} - \mu^{[k]} \nabla f(\mathbf{x}^{[k]}) \right)$$

- 更に $f = d_D^2/2$ とすると交互射影

⇒ 二乗距離は ι_D の Moreau エンベロープ

$$\mathbf{x}^{[k+1]} = P_C \left(P_D(\mathbf{x}^{[k]}) \right)$$

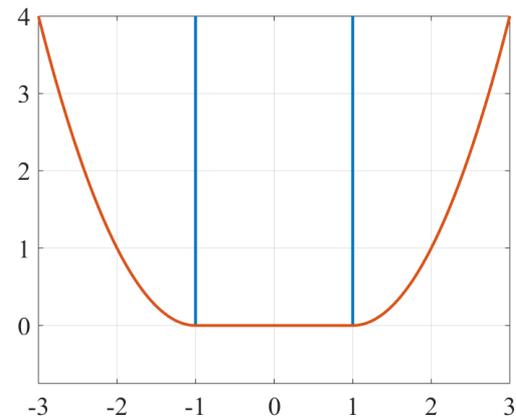
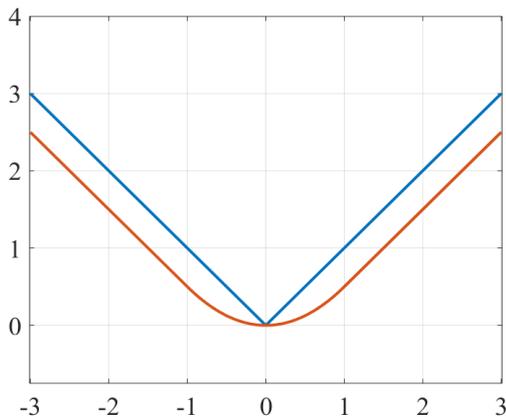
Moreau エンベロープはどのような近似か

凸関数を下から包み込むような近似

- 畳み込みのように滑らかにする効果がある
 - ⇒ **元が微分不可能な関数でも Moreau エンベロープは微分可能**
 - ⇒ しかも勾配は $(1/\mu)$ -Lipschitz 連続 (つまり勾配降下が収束する)

$$\nabla(\mu f) = (\text{Id} - \text{prox}_{\mu f})/\mu$$

- 絶対値関数 $|x|$ の Moreau エンベロープは huber 関数
- 指示関数 ι_C の Moreau エンベロープはその集合への距離関数 $d_C^2/2$



近接勾配法は微分不可能な項があっても良い

勾配法で扱いづらい項は近接作用素で対処

- 劣微分 ∂g は集合値関数だが近接作用素 $\text{prox}_g = (\text{Id} + \partial g)^{-1}$ は違う
⇒ 劣勾配法のように劣勾配（集合）の中から選ぶ必要がない
- 「微分不可能」や「勾配が Lipschitz 連続でない」項は近接作用素で
⇒ 微分可能かつ勾配が Lipschitz 連続な項はそのまま勾配降下で扱える

様々な実用的な関数に対して閉形式な解が存在

- 非負制約（マイナスの値を取らない）なら**距離射影**

$$\left[\text{prox}_{\iota_{\mathbb{R}_+}}(\mathbf{y}) \right]_n = \left[P_{\mathbb{R}_+}(\mathbf{y}) \right]_n = (y_n)_+ \quad \left(\cdot \right)_+ = \max\{\cdot, 0\}$$

- スパース性を誘導する ℓ_1 ノルムなら **Soft-Thresholding**

$$\left[\text{prox}_{\lambda \|\cdot\|_1}(\mathbf{y}) \right]_n = (1 - \lambda/|y_n|)_+ y_n$$

ステップ幅を制限した近接勾配法は補助関数法

近接勾配法の補助関数法としての解釈

- $f, g \in \Gamma_0(\mathcal{H})$, 勾配 ∇f が β -Lipschitz 連続
- $f(\mathbf{x}) + g(\mathbf{x})$ の上界として f を上から二次関数で押さえる
 $\Rightarrow \mu \in (0, 1/\beta]$ なら $f(\mathbf{x}) + g(\mathbf{x}) \leq \varphi(\mathbf{x}, \mathbf{y}) = \tilde{f}_\mu(\mathbf{x}, \mathbf{y}) + g(\mathbf{x})$

$$f(\mathbf{x}) \leq \tilde{f}_\mu(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \nabla f(\mathbf{y})^T (\mathbf{x} - \mathbf{y}) + \frac{1}{2\mu} \|\mathbf{x} - \mathbf{y}\|_2^2$$

- 上界関数 $\varphi(\cdot, \mathbf{x}^{[k]})$ の逐次最小化を行うのが**補助関数法**

$$\mathbf{x}^{[k+1]} = \arg \min_{\mathbf{x}} \varphi(\mathbf{x}, \mathbf{x}^{[k]})$$

- 右辺の部分問題の最適性条件 $\mathbf{0} \in \partial \varphi(\mathbf{x}, \mathbf{x}^{[k]}) = \nabla \tilde{f}_\mu(\mathbf{x}, \mathbf{x}^{[k]}) + \partial g(\mathbf{x})$
 \Rightarrow これを解いて上界を最小化すると $\mu \in (0, 1/\beta]$ の**近接勾配法**を得る

$$\mathbf{x}^{[k+1]} = \text{prox}_{\mu g}(\mathbf{x}^{[k]} - \mu \nabla f(\mathbf{x}^{[k]}))$$

ステップ幅を制限した近接勾配法は補助関数法

近接勾配法の補助関数法としての解釈

- $f, g \in \Gamma_0(\mathcal{H})$, 勾配 ∇f が β -Lipschitz 連続
- $f(\mathbf{x}) + g(\mathbf{x})$ の上界として f を上から二次関数で押さえる
 $\Rightarrow \mu \in (0, 1/\beta]$ なら $f(\mathbf{x}) + g(\mathbf{x}) \leq \varphi(\mathbf{x}, \mathbf{y}) = \tilde{f}_\mu(\mathbf{x}, \mathbf{y}) + g(\mathbf{x})$

$$f(\mathbf{x}) \leq \tilde{f}_\mu(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \nabla f(\mathbf{y})^T (\mathbf{x} - \mathbf{y}) + \frac{1}{2\mu} \|\mathbf{x} - \mathbf{y}\|_2^2$$

近接勾配法はステップ幅 $\mu \in (0, 2/\beta)$ で収束するので
 $1/\beta < \mu < 2/\beta$ のときは上界の最小点を通り過ぎている

- 右辺の部分問題の最適性条件 $\mathbf{0} \in \partial\varphi(\mathbf{x}^{[k]}, \mathbf{x}^{[k]}) = \nabla f_\mu(\mathbf{x}, \mathbf{x}^{[k]}) + \partial g(\mathbf{x})$
 \Rightarrow これを解いて上界を最小化すると $\mu \in (0, 1/\beta]$ の近接勾配法を得る

$$\mathbf{x}^{[k+1]} = \text{prox}_{\mu g}(\mathbf{x}^{[k]} - \mu \nabla f(\mathbf{x}^{[k]}))$$

不動点近似アルゴリズムとしての解釈

解集合と近接勾配作用素の不動点集合が一致

- 最適性条件より最適解 \mathbf{x}^* は **不動点集合** $\text{Fix}(\text{prox}_{\mu g}(\text{Id} - \mu \nabla f))$ の元

$$\mathbf{0} \in \nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*)$$

$$\mathbf{0} \in \mu \nabla f(\mathbf{x}^*) - \mathbf{x}^* + \mathbf{x}^* + \mu \partial g(\mathbf{x}^*)$$

$$(\text{Id} + \mu \partial g)(\mathbf{x}^*) \ni (\text{Id} - \mu \nabla f)(\mathbf{x}^*)$$

$$\mathbf{x}^* = (\text{Id} + \mu \partial g)^{-1}(\text{Id} - \mu \nabla f)(\mathbf{x}^*)$$

$$\mathbf{x}^* = \text{prox}_{\mu g}(\mathbf{x}^* - \mu \nabla f(\mathbf{x}^*))$$

- 近接勾配作用素は $\mu \in (0, 2/\beta)$ なら α -**平均非拡大写像** ($\alpha \in (0, 1)$)

$$\Rightarrow \text{prox}_{\mu g}(\text{Id} - \mu \nabla f) = (1 - \alpha)\text{Id} + \alpha S \text{ となる非拡大写像 } S \text{ が存在}$$

$$\Rightarrow \text{近接勾配作用素の場合は } \alpha = 2/(1 + 2 \min\{1, 1/(\mu\beta)\}) \in [2/3, 1)$$

- **Krasnosel'skii-Mann のアルゴリズム** (平均非拡大写像 T の反復適用)

$$\Rightarrow \mathbf{x}^{[k+1]} = T(\mathbf{x}^{[k]}) \text{ は } \text{Fix}(T) \text{ 内の1点に(弱)収束}$$

$$\Rightarrow \{\mathbf{x}^{[k]}\} \text{ は } \text{Fix}(T) \text{ に対し } \mathbf{Fejer monotone} \quad \|\mathbf{x}^* - \mathbf{x}^{[k+1]}\| \leq \|\mathbf{x}^* - \mathbf{x}^{[k]}\|$$

$$\forall \mathbf{x}^* \in \text{Fix}(T), \forall k \in \mathbb{N}$$

近接勾配アルゴリズムの単調性

コスト関数値も生成される点列も単調

■ 近接勾配法におけるコスト関数値の単調非増加性

⇒ $\mu \in (0, 2/\beta)$ の場合 $f(\mathbf{x}^{[k]}) + g(\mathbf{x}^{[k]})$ は上から押さえられる

$$\begin{aligned}(f + g)(\mathbf{x}^{[k+1]}) &\leq (f + g)(\mathbf{x}^{[k]}) - \frac{2 - \mu\beta}{2\mu} \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\|_2^2 \\ &\leq (f + g)(\mathbf{x}^{[k]}) - \frac{(2 - \mu\beta)\mu}{(1 + \mu\beta)^2} d_{\partial(f+g)(\mathbf{x}^{[k+1]})}(\mathbf{0})\end{aligned}$$

■ 近接勾配法における生成される点列の単調性

⇒ $\forall \mathbf{x}^* \in \arg \min_{\mathbf{x}} [f(\mathbf{x}) + g(\mathbf{x})]$ に対して距離が単調非増加

$$\|\mathbf{x}^* - \mathbf{x}^{[k+1]}\|_2^2 \leq \|\mathbf{x}^* - \mathbf{x}^{[k]}\|_2^2 - \frac{1 - \alpha}{\alpha} \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\|_2^2$$

近接勾配法の収束を加速させるテクニック

- 反復ごとに $\gamma^{[k]} \in [0, 1)$ を徐々に 1 に近づけていく
 - ⇒ 例えば $K > 0$ を定数として $\gamma_K^{[k]} = (k - 1)/(k + K - 1)$
 - ⇒ 収束レートを $O(1/k)$ から $O(1/k^2)$ に改善できる (最適)

$$\mathbf{y}^{[k+1]} = \mathbf{x}^{[k]} + \gamma^{[k]} (\mathbf{x}^{[k]} - \mathbf{x}^{[k-1]})$$

$$\mathbf{x}^{[k+1]} = \text{prox}_{\mu g} (\mathbf{y}^{[k+1]} - \mu \nabla f(\mathbf{y}^{[k+1]}))$$

- 二階の微分方程式を前進と後進 Euler で離散化した数値積分に対応
 - ⇒ 減衰係数を時間的に変化させることで高速に静止させる

$$\frac{d^2}{dt^2} \mathbf{x}(t) + \gamma(t) \frac{d}{dt} \mathbf{x}(t) + \nabla f(\mathbf{x}(t)) + \partial g(\mathbf{x}(t)) \ni \mathbf{0}$$

交互方向乗数法 (ADMM)

どちらの項も微分不可能な場合

- 近接勾配法は2つの関数のうち一方は微分可能であることを仮定
⇒ 微分できる項がなければ近接点アルゴリズムになる
- 微分不可能な関数の和を最小化するアルゴリズムも欲しい

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{Minimize}} \quad \mathcal{D}_d(\mathbf{x}) + \lambda \mathfrak{R}(\mathbf{x})$$

ADMM (Alternating Direction Method of Multiplier)

- **変数を分離**することでどちらの関数も近接作用素で扱う
⇒ 収束が速く非凸最適化問題にも有効なことが実験的に知られている

$$\mathbf{x}^{[k+1]} = \text{prox}_{\mathcal{D}_d} \left(\mathbf{z}^{[k]} - \mathbf{r}^{[k]} \right)$$

$$\mathbf{z}^{[k+1]} = \text{prox}_{\lambda \mathfrak{R}} \left(\mathbf{x}^{[k+1]} + \mathbf{r}^{[k]} \right)$$

$$\mathbf{r}^{[k+1]} = \mathbf{r}^{[k]} + \mathbf{x}^{[k+1]} - \mathbf{z}^{[k+1]}$$

主-双対分離法 (Primal-Dual Splitting method)

より複雑な問題を解けるアルゴリズムも多数存在

- 正則化項を複数用いることがよくある (複数の根拠を同時に利用)
⇒ 有界線形作用素と合成されていることもあって厄介

$$\text{Minimize}_{\mathbf{x} \in \mathcal{X}} \mathfrak{F}(\mathbf{x}) + \mathfrak{G}(L\mathbf{x}) + \mathfrak{H}(\mathbf{x})$$

各関数と行列を全て分離して扱えば簡単に

- 主-双対分離法では主問題と双対問題を同時に考えて解く
⇒ ADMMも主-双対分離法的一种と解釈できる

$$\mathbf{x}^{[k+1]} = \text{prox}_{\mu_1 \mathfrak{F}} \left(\mathbf{x}^{[k]} - \mu_1 (\nabla \mathfrak{H}(\mathbf{x}^{[k]}) + \mu_2 L^T \mathbf{z}^{[k]}) \right)$$

$$\mathbf{y} = \mathbf{z}^{[k]} + L(2\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]})$$

$$\mathbf{z}^{[k+1]} = \mathbf{y} - \text{prox}_{\frac{1}{\mu_2} \mathfrak{G}}(\mathbf{y})$$



信号処理と数理モデル

▶ 最適化アルゴリズム ◀

応用例





信号処理と数理モデル

最適化アルゴリズム

▶ 応用例 ◀



2次元位相のノイズ除去

- 干渉縞から光の位相を復元
 - ⇒ 高速度な計測はノイズが多い
 - ⇒ 位相画像のノイズ除去をしたい
- 先行研究：窓フーリエフィルタ
 - ⇒ スパースなスペクトログラム

複数の窓関数を利用

- スペクトログラムを複数用いる
 - ⇒ 狭い窓と広い窓で役割分担
- それらの極小畳み込みで正則化
 - ⇒ 適した窓の自動選択的なの？
- 加速付き線形化 ADMM で解いた

Convex optimization-based windowed Fourier filtering with multiple windows for wrapped-phase denoising

KOHEI YATABE* AND YASUHIRO OKAWA

Department of Intermedia Art and Science, Waseda University, 3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169-8555, Japan
*Corresponding author: k.yatabe@asagi.waseda.jp

Received 1 March 2016; revised 12 May 2016; accepted 15 May 2016; posted 16 May 2016 (Doc. ID 260304); published 7 June 2016

The windowed Fourier filtering (WFF), defined as a thresholding operation in the windowed Fourier transform (WFT) domain, is a successful method for denoising a phase map and analyzing a fringe pattern. However, it has some shortcomings, such as extremely high redundancy, which results in high computational cost, and difficulty in selecting an appropriate window size. In this paper, an extension of WFF for denoising a wrapped-phase map is proposed. It is formulated as a convex optimization problem using Gabor frames instead of WFT. Two Gabor frames with differently sized windows are used simultaneously so that the above-mentioned issues are resolved. In addition, a differential operator is combined with a Gabor frame in order to preserve discontinuity of the underlying phase map better. Some numerical experiments demonstrate that the proposed method is able to reconstruct a wrapped-phase map, even for a severely contaminated situation. © 2016 Optical Society of America

OCIS codes: (100.5070) Phase retrieval; (100.3175) Interferometric imaging; (100.3190) Inverse problems.

<http://dx.doi.org/10.1364/AO.55.004632>

1. INTRODUCTION

Interferometry-based measurement techniques generally observe information of interest as a wrapped-phase map. The so-called wrapped phase is a restriction of the original phase to the interval $[-\pi, \pi)$ by adding or subtracting an integer which is a multiple of 2π . The process of recovering the original phase from its wrapped version by removing the modulus 2π ambiguities, called phase unwrapping, is necessary to acquire the information. If the magnitude of phase difference between adjacent pixels is less than π (i.e., if the Itoh condition [1] is satisfied) the unwrapped phase can be obtained exactly by a path-following algorithm. However, the presence of measurement noise often violates the condition, and phase unwrapping becomes an ill-posed problem. Thus, many unwrapping techniques have been proposed during the last two decades to overcome such difficulties [2–8].

An effective strategy for eliminating the issue is to integrate a denoising filter before or within the unwrapping process [9–18]. Such a denoising process is usually desirable because the unwrapping problem becomes considerably easier as the noise level becomes lower. One successful approach is the windowed Fourier transform (WFT)-based filter, which is often called windowed Fourier filtering (WFF) [19–24]. The procedure of WFF is as follows. First, a wrapped-phase map φ is converted into the exponential phase field (EPF): $e^{i\varphi}$, where $i = \sqrt{-1}$. Second, EPF is represented by a linear combination

of windowed sinusoidal functions via WFT. Then, a thresholding operator is employed to enforce sparsity on the noisy EPF in the WFT domain. Finally, the inverse WFT provides denoised EPF, and the corresponding wrapped phase is obtained by calculating the principal value of its complex argument. Conversion of wrapped phase to EPF is necessary to remove the difficulty caused by the 2π discontinuity of the wrapping effect. Since EPF is a sinusoidal function, a Fourier-type transformation, including WFT, allows sparse representation of noiseless EPF. At the same time, noisy EPF is generally not sparse in the WFT domain. This sparsity-based characterization admits WFF to be a reasonable and effective approach for denoising the wrapped phase.

Although WFF has achieved great success, there are some shortcomings. The first point is high computational complexity. WFF is usually based on the full WFT, which results in extremely redundant representation. In general, redundancy in the transformed domain should be controlled properly in order to avoid unnecessary computation. Another point, which is closely related to the first one, is that WFF has less freedom on a choice of a window function. There are infinitely many pairs of a window function and its dual counterpart, which should be chosen depending on application. Nevertheless, the ordinary WFF seems to adopt only the Gaussian window, whose support is not compact. The last but most important point is that there is a trade-off between the effectiveness of

全変動は画像処理の定番

- 微分をスパースにする正則化
 - ⇒ 画像特有の区分定数性を誘導
- 音響の全変動の先行研究
 - ⇒ 複素スペクトログラムの正則化
 - ⇒ 位相を考慮したペナルティ

瞬時位相修正を提案

- 瞬時位相によるモデル誤差の低減
 - ⇒ 調波成分を無視した正則化
 - ⇒ 関数としての凸性は維持
- 時間領域でデータ忠実性を考慮
 - ⇒ 位相無矛盾なノイズ除去

PHASE CORRECTED TOTAL VARIATION FOR AUDIO SIGNALS

Kohei Yatabe and Yasuhiro Oikawa

Department of Intermedia Art and Science, Waseda University, Tokyo, Japan

ABSTRACT

In optimization-based signal processing, the so-called prior term models the desired signal, and therefore its design is the key factor to achieve a good performance. For audio signals, the time-directional total variation applied to a spectrogram in combination with phase correction has been proposed recently to model sinusoidal components of the signal. Although it is a promising prior, its applicability might be restricted to some extent because of the mismatch of the assumption to the signal. In this paper, based upon the previously proposed one, an improved prior for audio signals named instantaneous phase corrected total variation (iPCTV) is proposed. It can handle wider range of audio signals owing to the instantaneous phase correction term calculated from the observed signal.

Index Terms— Spectrogram, phase-aware processing, phase derivative, instantaneous frequency, convex optimization.

1. INTRODUCTION

For optimization-based signal processing methods, design of the so-called prior term, which imposes the prior knowledge about the desired signal, is important to achieve a good performance. While a complicated model is useful to obtain higher performance, a simple model is also important in practice. Recent methods may utilize several priors simultaneously to impose multiple aspects of the signal, and thus each prior is preferred to be simple as possible for reducing the overall complexity.

As such a simple prior term for audio signals, phase corrected total variation (PCTV) has been introduced in [1]. Total variation (composition of the first order difference and the l_1 -norm) is a quite popular prior, especially in image processing [2–5], that induces piece-wise smoothness to the signal. Based on the observation that a purely sinusoidal signal is represented by a smooth complex spectrogram after a suitable phase correction, PCTV is defined as the total variation applied time-directionally to the phase corrected spectrogram (see Section 2.1). Since it utilizes information of the phase spectrogram explicitly, PCTV can be regarded as a phase-aware prior, where such phase-aware methods receive much attention recently [6, 7]. It can easily be optimized by convex optimization techniques [8–12] because of its simplicity, and therefore PCTV is a promising prior which should be investigated further.

Although PCTV has many attractive properties, its applicability might be restricted to some extent because of the mismatch of the assumption to the signal. The key factor of PCTV is the phase correction which realizes the smooth time-frequency representation of a sinusoidal signal. However, the conventional PCTV deals with the phase in terms of the center frequency of each bin of the spectrogram. That is, the phase correction is performed not based on the phase of the signal but based on the parameters of a time-frequency analysis method. This mismatch might require a highly redundant time-frequency representation which may not be suitable for many applications of acoustical signal processing because of the computational complexity caused by the high redundancy.

In this paper, an improved PCTV, namely instantaneous phase corrected total variation (iPCTV), is proposed. It corrects the spectrogram based on the instantaneous phase of the signal so that the sinusoidal components are handled more appropriately. As the result of considering instantaneous phase, iPCTV can be applied to a broader range of spectrograms calculated by a larger shifting step of the window function than the conventional PCTV. For demonstrating its performance, a simple denoising problem is considered and is solved by the primal-dual splitting algorithm.

2. PHASE CORRECTED TOTAL VARIATION

In this section, after briefly reviewing the concept of the conventional PCTV, instantaneous correction of the phase based on the signal is proposed to improve the performance of PCTV.

2.1. The simple prior (Conventional PCTV) [1]

Let the short-time Fourier transform (STFT), or discrete Gabor transform, of a signal x with a window function w be defined as [13–15]

$$(\mathcal{F}^w x)(m, n) = \sum_{l=0}^{L-1} x(l+an) \overline{w(l) e^{2\pi i b m l / L}}, \quad (1)$$

where \bar{z} is complex conjugate of z , $i = \sqrt{-1}$, n and m are the time and frequency indices, and a and b are the time and frequency shifting steps, respectively. Since a sinusoidal signal (with initial phase ϕ_0) can be written as

$$s = e^{2\pi i (b f a n / L + \phi_0)} = e^{2\pi i (b f a (n+1) / L + \phi_0)} e^{-2\pi i b f a / L}, \quad (2)$$

its STFT has the neighborhood relation,

$$(\mathcal{F}^w s)(f, n+1) e^{-2\pi i b f a / L} = (\mathcal{F}^w s)(f, n), \quad (3)$$

when f coincides with some m . From this equation, it can be seen that the time-directional difference of the adjacent components of STFT is zero for a sinusoidal component of $f = m$ when the phase factor $e^{-2\pi i b f a / L}$ is multiplied.

In [1], the phase corrected version of STFT is considered,

$$(\mathcal{F}_{\text{PC}}^w x)(m, n) = (\mathcal{F}^w x)(m, n) e^{-2\pi i b m a n / L}, \quad (4)$$

which can be directly written as another form of STFT:

$$(\mathcal{F}_{\text{PC}}^w x)(m, n) = \sum_{l=0}^{L-1} x(l) \overline{w(l - an) e^{2\pi i b m a n / L}}. \quad (5)$$

Then, PCTV was defined as time-directional total variation of the phase corrected STFT¹:

$$\text{TV}_{\text{PC}}(x) = \|D_t \mathcal{F}_{\text{PC}}^w x\|_1 = \|D_t \mathcal{E}_{\text{PC}} \mathcal{F}^w x\|_1, \quad (6)$$

¹Note that this definition of PCTV is a reinterpreted version of the original description, where its motivation and description are detailed in [1].

ブラインド音源分離のための近接型アルゴリズム

独立性に基づく音源分離

- 負LogDet項とペナルティ項の和
 - ⇒ 多くの手法が共通の形式 (ICA, IVA, ILRMA など)
- まとめて解けたら便利はず
 - ⇒ ペナルティ項の差が手法の差
 - ⇒ より良いペナルティの探求

近接分離アルゴリズム

- ペナルティ項は近接作用素で扱う
 - ⇒ 1行の変更で多様な手法を実装
- 主-双対分離アルゴリズムを利用
 - ⇒ 非凸問題だけどちゃんと動く

DETERMINED BLIND SOURCE SEPARATION VIA PROXIMAL SPLITTING ALGORITHM

Kohei Yatabe[†] and Daichi Kitamura^{‡*}

[†]Department of Intermedia Art and Science, Waseda University, Tokyo, Japan

[‡]Department of Information Physics and Computing, The University of Tokyo, Tokyo, Japan

ABSTRACT

The state-of-the-art algorithms of determined blind source separation (BSS) methods based on the independent component analysis (ICA) have gained computational efficiency by the majorization-minimization (MM) principle with a price of losing flexibility. That is, replacing and comparing different source models are not easy in such MM-based framework because it requires efforts to derive a new algorithm each time when one changes the model. In this paper, a general framework for obtaining an ICA-based BSS algorithm is proposed so that a source model can easily be replaced because only a single line of the algorithm must be modified. A sparsity-based extension of the independent vector analysis and a low-rankness-based BSS model using the nuclear norm are also proposed to demonstrate the simplicity and easiness of the proposed framework.

Index Terms— Independence-based separation, frequency domain independent component analysis (FDICA), independent vector analysis (IVA), primal-dual splitting algorithm, proximity operator.

1. INTRODUCTION

Blind source separation (BSS) is methodology for recovering source signals from multiple mixtures without any knowledge about the mixing system. Let a convolutive mixing process be approximated in time-frequency domain as

$$\mathbf{x}[t, f] \approx A[f] \mathbf{s}[t, f], \quad (1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_M]^T$ is an observation obtained by M microphones, $\mathbf{s} = [s_1, s_2, \dots, s_N]^T$ is a source signal to be recovered, $A[f]$ is an $M \times N$ mixing matrix, and t and f are indices of time and frequency, respectively. Then, the aim of BSS is to recover N source signals \mathbf{s} from the mixtures \mathbf{x} . In a determined or overdetermined situation ($M \geq N$), many of the BSS problems are formulated as an estimation problem of finding an $N \times M$ demixing matrix $W[f]$ which is a left inverse of $A[f]$ (i.e., $W[f]A[f] = I$), and the source signals are recovered by simple multiplication:

$$W[f] \mathbf{x}[t, f] \approx W[f] A[f] \mathbf{s}[t, f] = \mathbf{s}[t, f]. \quad (2)$$

For the sake of simplicity, only a determined situation ($M = N$) is considered in this paper.

For estimating a demixing matrix $W[f]$, statistical independence between source signals is often assumed that leads to a family of independence-based BSS algorithms. Arguably, independent component analysis (ICA) [1] applied in frequency domain (FDICA) [2–6] is one of the most famous methods among them. However, FDICA suffers from the so-called permutation problem [7–10], and thus some recent developments on BSS aim to

avoid it by considering more sophisticated models of source signals. For instance, independent vector analysis (IVA) [11–13] assumes co-occurrence among the frequency components in each source, and independent low-rank matrix analysis (ILRMA) [14–16] assumes low-rankness on spectrogram of each source. The key to success of these methods is to incorporate prior knowledge of source signals into their formulations. That is, improvement brought by these methods relies on the preciseness of their source models. Therefore, seeking a better model is the important process for developing a novel and effective BSS method.

However, recent algorithms [16–18] cannot be applied to a different source model directly because they are specialized to each method. These state-of-the-art algorithms are based on the majorization-minimization (MM) principle which requires specially designed upper-bound of the objective functions. That is, one has to derive a new algorithm each time as the source model is modified. Therefore, it may take a lot of time to examine a new source model especially when the model is a complicated one. If a single algorithm can handle a large number of source models without effort, discovering a better source model should become much easier that possibly boosts the development of BSS.

In this paper, a flexible framework for independence-based BSS is proposed based on a proximal splitting algorithm [19–22]. The usefulness of the splitting algorithm comes from its capability of splitting an optimization problem into several easier subproblems which are handled by the proximity operators. We take advantage of this feature to split the ICA-based BSS problem into two parts so that different source models can easily be combined by modifying only a single line of the algorithm. The proposed framework is tested by introducing three new source models (a sparsity-based extension of IVA, a low-rankness-based model using the nuclear norm, and a sparsity-based extension of the low-rank model), and potentiality of the proposed method is indicated by the result.

2. INDEPENDENCE-BASED BSS

As introduced in the previous section, independence-based BSS methods aim to estimate $M \times M$ demixing matrices $\{W[f]\}_{f=1}^F$ which approximately recover the source signals from the observations as $W[f] \mathbf{x}[t, f] \approx \mathbf{s}[t, f]$. Many of them fall into a minimization problem of the following form:

$$\underset{\{W[f]\}_{f=1}^F}{\text{Minimize}} \mathcal{P}(W[f] \mathbf{x}[t, f]) - \sum_{f=1}^F \log |\det(W[f])|, \quad (3)$$

where \mathcal{P} is a real-valued penalty function corresponding to the source model. For example, with some constant C ,

$$\mathcal{P}(\mathbf{y}[t, f]) = C \|\mathbf{y}[t, f]\|_1 = C \sum_{m=1}^M \sum_{n=1}^N |y_m[t, f]| \quad (4)$$

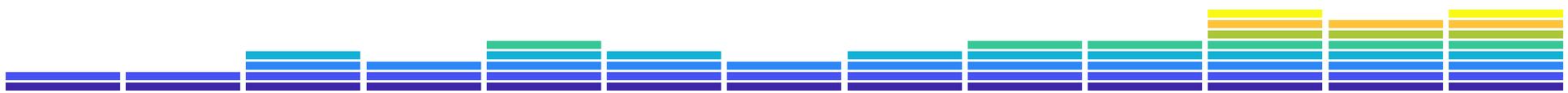
*This work was partly supported by JSPS Grant-in-Aid for Research Activity Start-up (17H06572, 17H07191).



1-4-10

近接分離最適化による ブラインド音源分離

○ 矢田部 浩平（早大）, 北村 大地（東大）



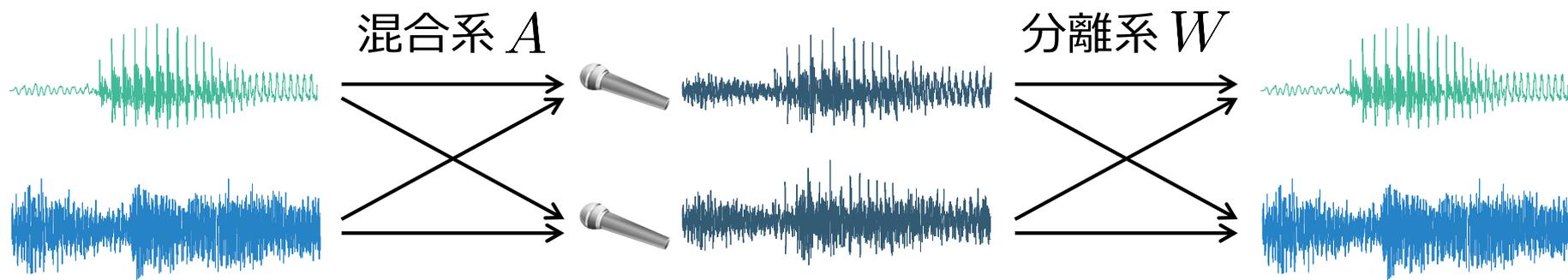
ブラインド音源分離問題

混合された信号を混合前の信号に分離

- マイク位置や音源位置などの事前情報を用いずに推定（ブラインド）

独立成分分析の発展形が盛んに研究されている

- 混合系の逆（分離系）を推定することで信号を復元
 - **独立性基準**によるブラインド音源分離
- ⇒ 統計的な独立性を最大化するように分離フィルタを推定する



独立性基準によるブラインド音源分離問題

音源モデル項と負Logdet項の最小化問題に帰着

- 負Logdet項は各分離手法で共通（自明解を避ける効果）
⇒ 分離手法ごとの違いは**音源モデル項の違い**として現れる

$$\text{Minimize}_{\{W[f]\}_{f=1}^F} \underbrace{\mathcal{P}(W[f]\mathbf{x}[t, f])}_{\text{音源モデル項}} - \sum_{f=1}^F \log |\det(W[f])|$$

新たな音源モデルが新たな手法を生む

- より**良い音源モデルの発見**によって**性能が向上できる**と期待される
⇒ 様々なモデルを試行錯誤することで手法が改良・改善されてきた

分離性能は音源モデル項によって決まる

音源モデル項の違いが分離結果の違いを与える

$$\text{Minimize}_{\{W[f]\}_{f=1}^F} \mathcal{P}(W[f]\mathbf{x}[t, f]) - \sum_{f=1}^F \log |\det(W[f])|$$

- Laplace 分布に基づく **周波数領域独立成分分析 (FDICA)**

$$\mathcal{P}(\mathbf{y}[t, f]) = C \|\mathbf{y}[t, f]\|_1 = C \sum_{m=1}^M \sum_{t=1}^T \sum_{f=1}^F |y_m[t, f]|$$

- 球対称 Laplace 分布に基づく **独立ベクトル分析 (IVA)**

$$\mathcal{P}(\mathbf{y}[t, f]) = C \|\mathbf{y}[t, f]\|_{2,1} = C \sum_{m=1}^M \sum_{t=1}^T \left(\sum_{f=1}^F |y_m[t, f]|^2 \right)^{\frac{1}{2}}$$

- 板倉斎藤 NMF に基づく **独立低ランク行列分析 (ILRMA)**

$$\mathcal{P}(\mathbf{y}[t, f]) = C \sum_{m=1}^M \min_{\varphi_{f,r}^{[m]} \geq 0, \psi_{r,t}^{[m]} \geq 0} \sum_{t=1}^T \sum_{f=1}^F \left(\frac{|y_m[t, f]|^2}{\sum_{r=1}^R \varphi_{f,r}^{[m]} \psi_{r,t}^{[m]}} + \log \sum_{r=1}^R \varphi_{f,r}^{[m]} \psi_{r,t}^{[m]} \right)$$

音源分離における最適化アルゴリズム

最適化問題を解いた結果が分離結果

- 多くの最適化問題は反復アルゴリズムを用いて解く必要がある
⇒ 一般的なアルゴリズム vs 特別設計されたアルゴリズム

勾配降下法 (gradient descent method)

- 勾配を利用しコスト関数が小さくなる方向に反復更新
 - 微分さえできればアルゴリズムを導出可能 (簡単)
 - × 微分できないと困る, パラメータ調整が容易でない, 収束が遅い

補助関数法 (majorization-minimization method)

- コスト関数に合わせて補助関数を特別に設計して逐次最小化
 - 収束が速い, パラメータ調整が不要
 - △ コスト関数に適した補助関数を見出す必要, 不安定な演算を含む

より一般的で適用範囲の広いアルゴリズムの提案

- 多くの音源分離モデルを**統一的に扱えるアルゴリズム**
 - ⇒ 微分不可能だったり非有限なモデルでも緩和問題として対応可能
- 音源モデルを取り換えた際の**導出が簡単なアルゴリズム**
 - ⇒ 負Logdet項を分離することで音源モデル項のみ考えれば良くなる
- 観測信号の条件によらない**安定的なアルゴリズム**
 - ⇒ 逆行列演算などの条件によって不安定になる操作を含まない

提案手法：主双対分離型の近接アルゴリズム

- 近接分離アルゴリズムを独立性基準の音源分離問題に適用
 - ⇒ アルゴリズムが適用できるように変形して問題自体を新たに定義
- 複数の項からなる音源モデルを扱うことが可能

今回扱う目的関数

負Logdet項と何らかの関数の和の最小化

- 負Logdet項は手法（音源モデル）によらず共通
 - ⇒ **負Logdet項と音源モデル項を別々に扱えるアルゴリズム**が好ましい
 - ∴ 音源項のみを考えてアルゴリズムを変形すれば良い

$$\text{Minimize}_{\{W[f]\}_{f=1}^F} \underbrace{\mathcal{P}(W[f]\mathbf{x}[t, f])}_{\text{モデル毎に変化}} - \sum_{f=1}^F \underbrace{\log |\det(W[f])|}_{\text{共通}}$$

主・双対分離アルゴリズムを適用

- 近年注目されている凸最適化アルゴリズムの1つ
 - ⇒ 目的関数の複数の項を**手続きとして分離して解く**ことで効率化

主双対分離アルゴリズム (Primal-dual splitting)

複数の項の和の最小化を行う反復解法

- 関数の和で定義される問題をより簡単な部分問題に分解して解く
- 有界線形作用素との合成関数を含む場合も効率良く解くことが可能

$$\underset{\mathbf{w}}{\text{Minimize}} \quad g(\mathbf{w}) + h(L\mathbf{w})$$

$$\left[\begin{array}{l} \tilde{\mathbf{w}} = \text{prox}_{\mu_1 g} \left[\mathbf{w}^{[k]} - \mu_1 \mu_2 L^T \mathbf{y}^{[k]} \right] \\ \mathbf{z} = \mathbf{y}^{[k]} + L(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]}) \\ \tilde{\mathbf{y}} = \mathbf{z} - \text{prox}_{h/\mu_2} \left[\mathbf{z} \right] \\ (\mathbf{w}^{[k+1]}, \mathbf{y}^{[k+1]}) = \alpha(\tilde{\mathbf{w}}, \tilde{\mathbf{y}}) + (1 - \alpha)(\mathbf{w}^{[k]}, \mathbf{y}^{[k]}) \end{array} \right.$$

近接作用素 (proximity operator)

近接項により扱いやすくなった部分問題

- その関数を**現在地の近傍で最小化**する作用素
 - ⇒ g が微分可能で μ が小さいときは勾配法の 1 ステップに似た操作
- 微分不可能な関数でも**微分可能な緩和問題**になってるので解くのが楽
 - ⇒ 2 ノルムとの極小畳み込み (Moreauエンベロップ)

$$\text{prox}_{\mu g}[\mathbf{y}] = \arg \min_{\mathbf{z}} \left[g(\mathbf{z}) + \frac{1}{2\mu} \|\mathbf{y} - \mathbf{z}\|_2^2 \right]$$

多くの有用な関数の近接作用素は解析解が存在

- 解析解を代入することで直ちにアルゴリズムを導出可能
- 解析解がない場合でも反復解法を適用することで対処可能
 - ⇒ アルゴリズムの一部を差し替えるのみで音源モデルをテスト可能

Logdet項の近接作用素

Logdet項を特異値で緩和

- 特異値で書き換えることでユニタリ不変関数に

$$\mathcal{I}(W) = - \sum_{f=1}^F \sum_{m=1}^M \log |\sigma_m(W[f])|$$

各特異値に負Logの近接作用素を適用すれば良い

- それぞれの特異値に少しだけ値を足す操作（安定的に計算可）

$$\text{prox}_{\mu \mathcal{I}}[W] = U \text{prox}_{-\mu \log}[\Sigma(W)] V^T$$

$$\left(\text{prox}_{-\mu \log}[\Sigma] \right)_{mm} = \frac{\sigma_m + \sqrt{\sigma_m^2 + 4\mu}}{2}$$

音源モデル項の解釈

観測データによって決まる有界線形作用素との合成

- 最適化変数は分離フィルタなので**観測データは分離フィルタの変換**
⇒ 分離フィルタをベクトル化して書けば観測データは行列

$$\mathcal{P}(W[f]\mathbf{x}[t, f]) \quad \longrightarrow \quad \mathcal{P}(X\mathbf{w})$$

分離フィルタ推定問題のベクトル表現

- 主双対分離型アルゴリズムが適用できる形式
⇒ Logdet項の近接作用素は前述の通り
⇒ 音源モデル項の近接作用素のみを気にすれば良い

$$\underset{\mathbf{w}}{\text{Minimize}} \quad \mathcal{I}(\mathbf{w}) + \mathcal{P}(X\mathbf{w})$$

Algorithm 1 PDS-BSS

- 1: **Input:** $X, \mathbf{w}^{[1]}, \mathbf{y}^{[1]}, \mu_1, \mu_2, \alpha$
 - 2: **Output:** $\mathbf{w}^{[K+1]}$
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 X^H \mathbf{y}^{[k]}]$
 - 5: $\mathbf{z} = \mathbf{y}^{[k]} + X(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$
 - 6: $\tilde{\mathbf{y}} = \mathbf{z} - \text{prox}_{\frac{1}{\mu_2} \mathcal{P}}[\mathbf{z}]$
 - 7: $\mathbf{y}^{[k+1]} = \alpha \tilde{\mathbf{y}} + (1 - \alpha) \mathbf{y}^{[k]}$
 - 8: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$
 - 9: **end for**
-

Algorithm 1 PDS-BSS

1: **Input:** $X, \mathbf{w}^{[1]}, \mathbf{y}^{[1]}, \mu_1, \mu_2, \alpha$

2: **Output:** $\mathbf{w}^{[K+1]}$

3: **for** $k = 1, \dots, K$ **do**

4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 X^H \mathbf{y}^{[k]}]$

5: $\mathbf{z} = \mathbf{y}^{[k]} + X(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$

6: $\tilde{\mathbf{y}} = \mathbf{z} - \text{prox}_{\frac{1}{\mu_2} \mathcal{P}}[\mathbf{z}]$

7: $\mathbf{y}^{[k+1]} = \alpha \tilde{\mathbf{y}} + (1 - \alpha) \mathbf{y}^{[k]}$

8: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$

9: **end for**

特異値分解をして
特異値を増やす操作

音源項の近接作用素
(音源項のみの最適化)

Algorithm 2 PDS-BSS-multiPenalty

- 1: **Input:** $X, \mathbf{w}^{[1]}, \mathbf{y}_1^{[1]}, \dots, \mathbf{y}_Q^{[1]}, \mu_1, \mu_2, \alpha$
 - 2: **Output:** $\mathbf{w}^{[K+1]}$
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 X^H (\sum_{q=1}^Q \mathbf{y}_q^{[k]})]$
 - 5: **for** $q = 1, \dots, Q$ **do**
 - 6: $\mathbf{z}_q = \mathbf{y}_q^{[k]} + X(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$
 - 7: $\tilde{\mathbf{y}}_q = \mathbf{z}_q - \text{prox}_{\frac{1}{\mu_2} \mathcal{P}_q} [\mathbf{z}_q]$
 - 8: $\mathbf{y}_q^{[k+1]} = \alpha \tilde{\mathbf{y}}_q + (1 - \alpha) \mathbf{y}_q^{[k]}$
 - 9: **end for**
 - 10: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$
 - 11: **end for**
-

Algorithm 2 PDS-BSS-multiPenalty

- 1: **Input:** $X, \mathbf{w}^{[1]}, \mathbf{y}_1^{[1]}, \dots, \mathbf{y}_Q^{[1]}, \mu_1, \mu_2, \alpha$
 - 2: **Output:** $\mathbf{w}^{[K+1]}$
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 X^H (\sum_{q=1}^Q \mathbf{y}_q^{[k]})]$
 - 5: **for** $q = 1, \dots, Q$ **do**
 - 6: $\mathbf{z}_q = \mathbf{y}_q^{[k]} + X(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$
 - 7: $\tilde{\mathbf{y}}_q = \mathbf{z}_q - \text{prox}_{\frac{1}{\mu_2} \mathcal{P}_q} [\mathbf{z}_q]$
 - 8: $\mathbf{y}_q^{[k+1]} = \alpha \tilde{\mathbf{y}}_q + (1 - \alpha) \mathbf{y}_q^{[k]}$
 - 9: **end for**
 - 10: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$
 - 11: **end for**
-

特異値拡大

音源項の近接作用素
(各項を独立に解く)

提案アルゴリズム

$$\text{Minimize}_{\mathbf{w}} \mathcal{I}(\mathbf{w}) + \sum_{q=1}^Q \mathcal{P}_q(X\mathbf{w})$$

$0 < \alpha < 2$
なら何でも良い

Algorithm 2 PDS-BSS-multiPenalty

- 1: **Input:** $X, \mathbf{w}^{[1]}, \mathbf{y}_1^{[1]}, \dots, \mathbf{y}_Q^{[1]}, \mu_1, \mu_2, \alpha$
- 2: **Output:** $\mathbf{w}^{[K+1]}$
- 3: **for** $k =$ **ゼロで良い**
- 4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 X^T \mathbf{y}_q^{[k]}]$
- 5: **for** $q = 1, \dots, Q$ **do**
- 6: $\mathbf{z}_q = \mathbf{y}_q^{[k]} + X(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$
- 7: $\tilde{\mathbf{y}}_q = \mathbf{z}_q - \text{prox}_{\frac{1}{\mu_2} \mathcal{P}_q}[\mathbf{z}_q]$
- 8: $\mathbf{y}_q^{[k+1]} = \alpha \tilde{\mathbf{y}}_q + (1 - \alpha) \mathbf{y}_q^{[k]}$
- 9: **end for**
- 10: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$
- 11: **end for**

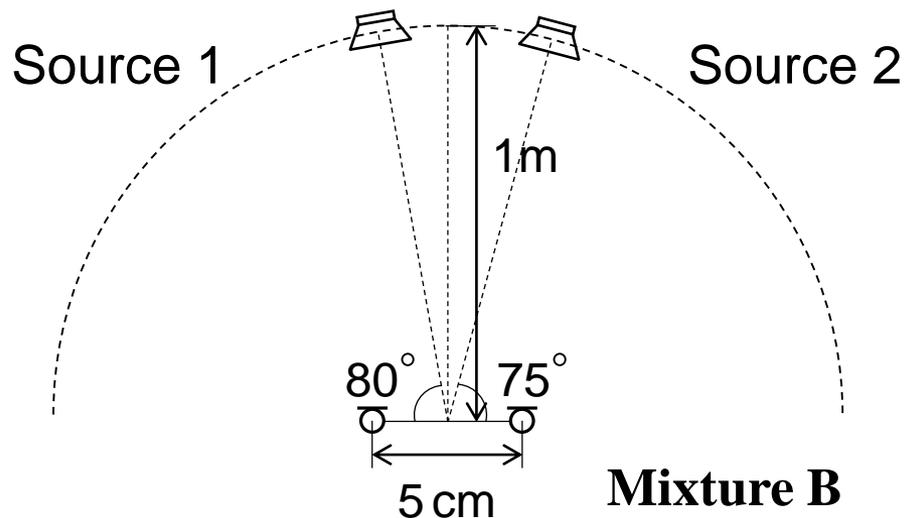
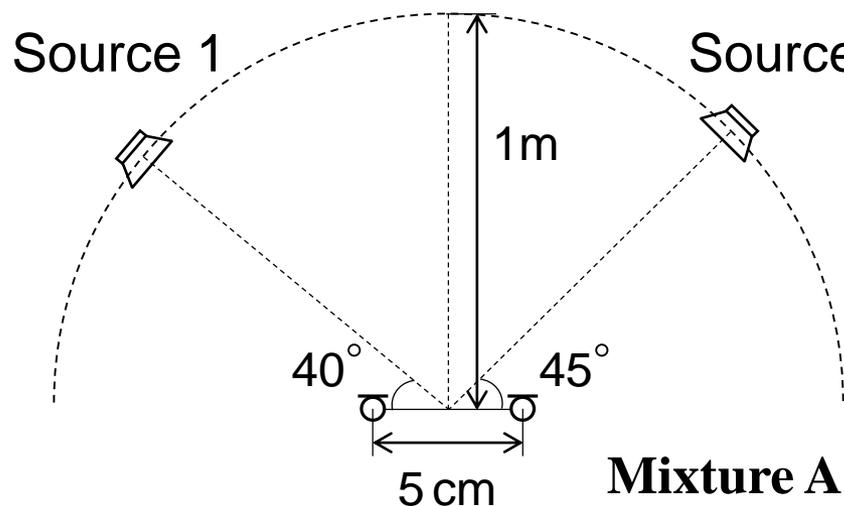
提案する正規化法

$$X \leftarrow X / (\sqrt{Q} \|X\|_s)$$

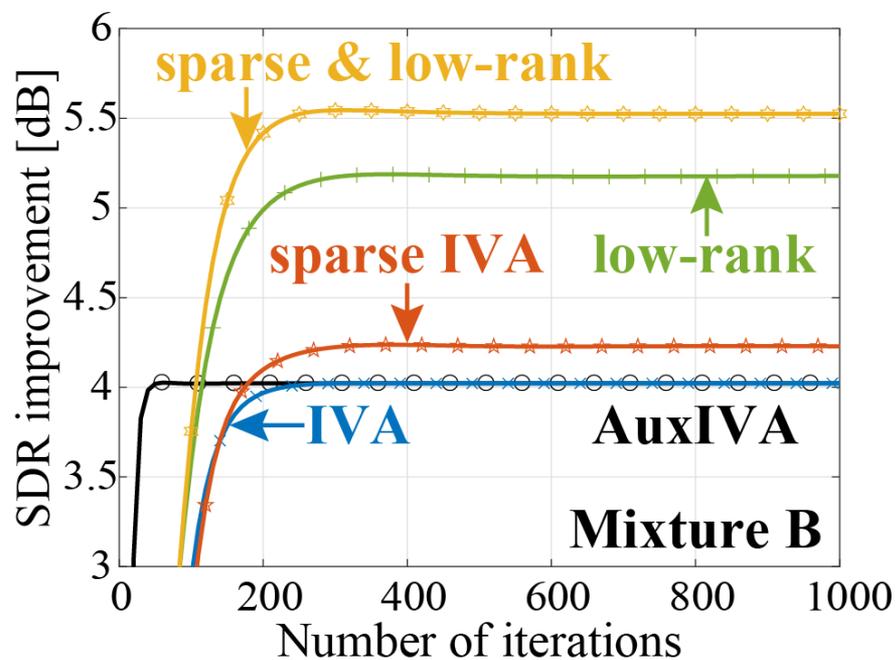
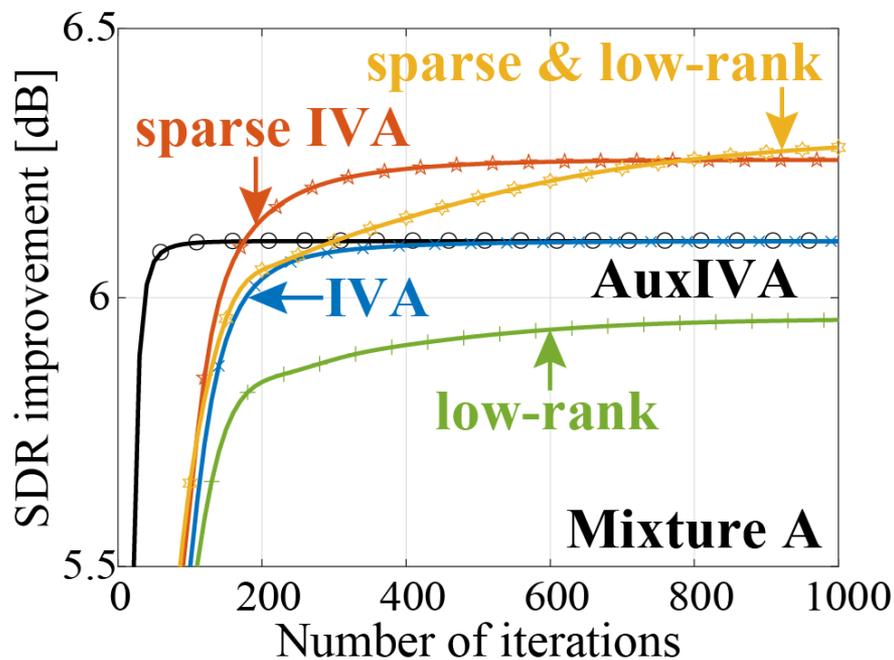
を適用すれば
どちらも1で良い

SiSEC 2011 (UND, dev1) liverec [Araki et al., 2012]

- SiSEC 2011の**実環境収録音源**を使用
 - ⇒ 4音源2チャンネル収録音源を2つに分けて使用
- 残響時間：130ミリ秒
- ハン窓（128ミリ秒，ハーフオーバーラップ）
- 分離フィルタは単位行列で初期化



実験結果 (SDR改善量)



IVA

$$\mathcal{I}(\mathbf{w}) + \|X\mathbf{w}\|_{2,1}$$

sparse IVA

$$\mathcal{I}(\mathbf{w}) + \|X\mathbf{w}\|_{2,1} + \lambda \|X\mathbf{w}\|_1$$

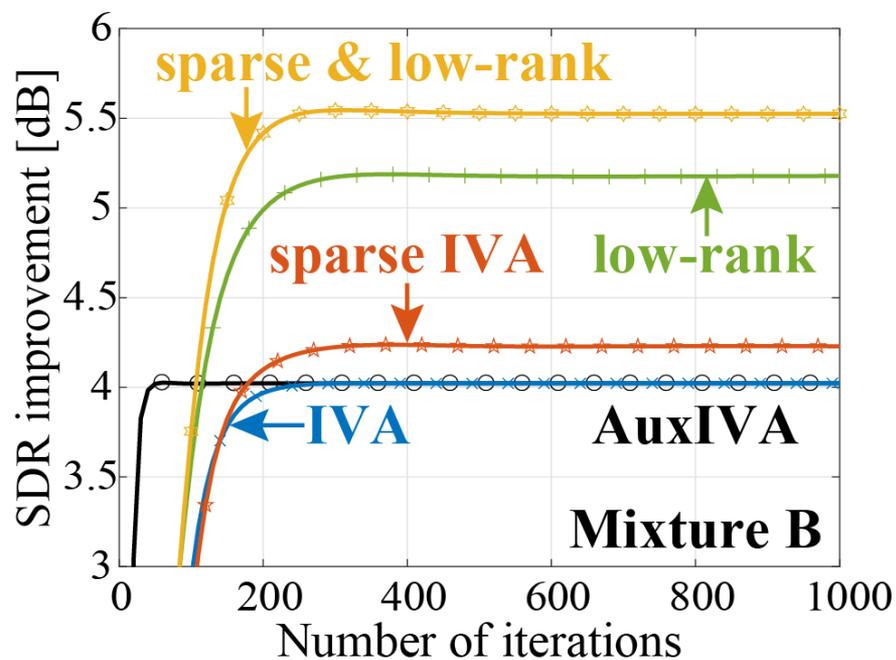
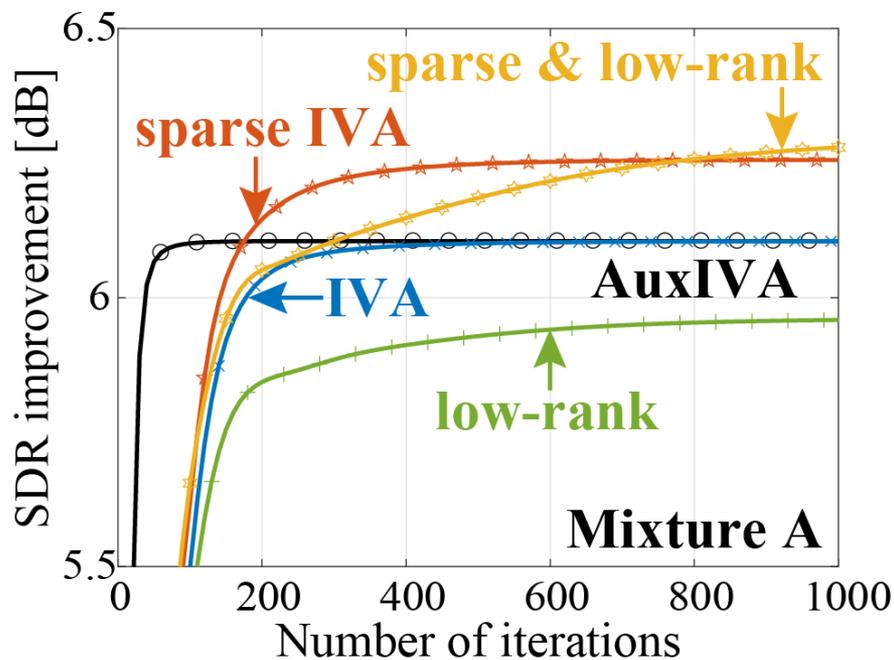
low-rank

$$\mathcal{I}(\mathbf{w}) + \|\mathcal{M}_*(X\mathbf{w})\|_*$$

sparse & low-rank

$$\mathcal{I}(\mathbf{w}) + \|\mathcal{M}_*(X\mathbf{w})\|_* + \lambda \|X\mathbf{w}\|_1$$

実験結果 (SDR改善量)



IVA

$$\mathcal{I}(\mathbf{w}) + \|X\mathbf{w}\|_{2,1}$$

sparse IVA

$$\mathcal{I}(\mathbf{w}) + \|X\mathbf{w}\|_{2,1} + \lambda \|X\mathbf{w}\|_1$$

反復 1 回あたりの計算時間
(Core i5-7200U, MATLAB 2017a)

AuxIVA : 80.8 ms
提案手法 : 47.4 ms