

深層展開を用いた行列補完手法の高速化

著者	内藤 凜
出版者	法政大学大学院情報科学研究科
雑誌名	法政大学大学院紀要. 情報科学研究科編
巻	18
ページ	1-4
発行年	2023-03-24
URL	http://doi.org/10.15002/00026289

深層展開を用いた行列補完手法の高速化

An Acceleration Technique for Matrix Completion using Deep Unfolding

内藤凜*

Rin Naito

法政大学大学院 情報科学研究科 情報科学専攻

Email: rin.naito.3n@stu.hosei.ac.jp

Abstract—This paper proposes an acceleration technique using deep unfolding for matrix completion problem, which is the problem of estimating the missing values of a matrix. To solve the problem, various methods have been proposed, which is based on matrix rank minimization. While these methods have iterative schemes to minimize the matrix rank on each step, they have poor estimation accuracy and convergence property when the parameters of the iterative schemes are not given appropriately. In order to provide appropriate parameters of iterative schemes, this paper focuses on a deep unfolding to accelerate matrix completion methods. The deep unfolding is a method of constructing more flexible derivation algorithms by embedding learnable parameters into existing iterative algorithms and by learning the parameters applying deep learning techniques to differentiable iterative algorithms with inputs and outputs. Algorithms which have adjustable parameters such as gradient descent are expected to accelerate by using deep unfolding. This paper proposes two matrix completion methods accelerated by deep unfolding which can optimize the iterative parameters via machine learning. Numerical experiments show the effectiveness of the proposed methods.

1. はじめに

本論文では、行列補完問題に対する既存のアルゴリズムの高速化について扱う。行列補完問題とは行列の一部の要素を基に未知の要素を推定し、欠損値を補完する問題の事である。同問題に対する一般的な解法として、行列の低ランク化に基づく手法が数多く存在する [1], [2], [3], [4]。行列ランク最小化問題は組み合わせ最適化問題の性質を持っているため、一般的には NP 困難な問題である。そのためランクの代わりに別の目的関数を用い、逐次的に近似解を得る手法が提案されているが、計算量が多くまた適切な内部パラメータの設定が困難であることから、短時間で解を求めることは難しい [1]。また、多くの既存のアルゴリズムではその求解に勾配降下法を用いており、そのステップサイズに補完性能が依存する。

本論文では行列補完手法の高速化を実現するために深層展開 [5] に着目した。深層展開とは深層学習技術を入出力を伴う微分可能な反復型アルゴリズムに対して適用出来ることを応用して、既存の反復型アルゴリズムに対して学習可能なパラメータを埋め込み、そのパラメータを学習することによってより柔軟な派生アルゴリズムを構成する手法のことである。深層展開を用いることで勾配降下法などの従来の知見に基づくアルゴリズムの高速化が見込まれている。そこで、本論文では交互勾配降下法 (Alternating Gradient Descent, 以下 AGD と称する) を用いた行列補完として NSAO (Null Space based

Alternating Optimization) 法 [1], [3], [4], [6] と行列因子分解に基づく行列補完 [2], [7] に着目し、それらに対して深層展開を適用することで収束加速を実現する手法を提案する。

2. 準備

2.1. 行列補完問題

本節では、一般的な行列補完問題について説明する。行列補完問題とは、対象となる欠損行列 $X \in \mathbb{R}^{m \times n}$ ($m < n$) の内の観測可能な要素を基に欠損部分を推定する問題の事である。この問題は以下の式で示される。

$$\text{Find } X \text{ s.t. } P_{\Omega}(X) = P_{\Omega}(X^*) \quad (1)$$

このとき、 Ω は与えられた要素を示す添字集合、 $P_{\Omega}: \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ は集合 Ω に含まれる添字以外の全ての要素を 0 に変換する線形作用素、 X^* は真の行列を示している。この行列補完問題の古典的解法として、以下のような行列ランク最小化問題が提案されている。

$$\begin{aligned} &\text{Minimize} && \text{rank } X \\ &\text{subject to} && P_{\Omega}(X) = P_{\Omega}(X^*) \end{aligned} \quad (2)$$

この問題は制約条件である $P_{\Omega}(X) = P_{\Omega}(X^*)$ により与えられた要素を変換させることなく最小のランクとなる行列 X を求める問題だが、一般的には組み合わせ最適化としての性質を持つため NP 困難な問題である。そこで、この問題を置き換えて近似解を得るための手法がいくつか提案されている。

2.2. 深層展開 (Deep Unfolding)

深層学習技術は、人間の神経細胞の仕組みを模したニューラルネットワークを用いて人間が行うタスクをコンピュータに学習させる機械学習の手法の一つであり、画像・音声認識や医療研究、自動翻訳といった多種多様な分野において活用されている技術である。通常の深層学習ではニューラルネットワークに対して学習を行い、入力されたデータの分類や識別器の構成などを行う。しかし近年、ニューラルネットワーク以外のアルゴリズムに対して深層学習技術を適用する技術が研究されている。アルゴリズムの内部プロセスが微分可能である場合、深層学習技術をそのアルゴリズム内部に埋め込んだ学習可能パラメータに対して適用することによって、与えられた教師データに基づいてパラメータを学習させることが出来る。

一般的な反復構造をもつアルゴリズムの流れを図 1 に示す。入力は図 1 の左側から入り、プロセス A,B,C による処理を終えた後にその出力を再度入力として処理を行う。このとき、プロセス A,B,C の非線形関数が微分可能であり、かつ非ゼロの導関数値をもつ多変数関数であるならば、深層学習技術をこの反復アルゴリズムに適

* Supervisor: Prof. Konishi Katsumi

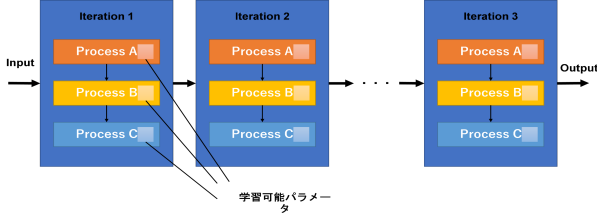


図 1. 反復型アルゴリズムの概略図

用することが可能である。すなわちアルゴリズムの内部に学習可能なパラメータを埋め込み、そのパラメータに対して学習を行う。このように、反復型アルゴリズムを対象として深層学習技術を適用するアプローチのことを深層展開と呼ぶ。深層展開を適用するメリットとして、従来の知見に基づいて設計された勾配法や射影勾配法といった反復アルゴリズムに対して深層展開を適用することで、より柔軟なアルゴリズムを生成できる点が挙げられる。また、ニューラルネットワークを用いた深層学習に対して学習可能パラメータの数が非常に少ないという特徴を持つため、アルゴリズムにおけるそれぞれの学習可能パラメータの役割を把握しやすく、従来手法との比較が行いやすいことも長所として挙げられる。

単純な勾配降下法に対して深層展開を適用した例を示す。まず、以下の式で与えられる更新則を考える。

$$s_{t+1} = s_t - \gamma \nabla f(s_t) \quad (t = 0, 1, \dots, T-1) \quad (3)$$

このとき、 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ は目的関数を、 γ は固定値でありステップサイズパラメータを示す。この勾配降下法に対して、 γ を学習可能パラメータとして深層展開を適用した場合、勾配ステップは以下のように定義される。

$$s_{t+1} := s_t - \gamma_t \nabla f(s_t) \quad (4)$$

この勾配ステップを持つ勾配法を DUGD (Deep Unfolding Gradient Descent) 法と呼ぶ [5]。

3. 提案手法

本章では、行列ランク最小化に基づく行列補完手法である NSAO 法 [1], [3], [4], [6] と行列因子分解 [2], [7] に基づく行列補完に使用される AGD に対し、深層展開を適用する提案手法について説明をする。

3.1. NSAO 法に基づく行列補完

はじめに、対象となる行列 X のランクが低いと仮定した時、 $X = WX$ を満たすような低ランク行列 $W \in \mathbb{R}^{m \times m}$ が存在する。この考えに基づいて、低ランクの行列補完問題は以下のように定義される。

$$\begin{aligned} & \text{Minimize} && \|W\|_F^2 \\ & \text{subject to} && X = WX, P_\Omega(X) = P_\Omega(X^*) \end{aligned} \quad (5)$$

このとき、 $\|\cdot\|_F$ は行列のフロベニウスノルムを表す。文献 [1] から式 (5) は以下のように緩和される。

$$\begin{aligned} & \text{Minimize} && \|(I - W)X\|_F^2 + \gamma \|W\|_F^2 \\ & \text{subject to} && P_\Omega(X) = P_\Omega(X^*) \end{aligned} \quad (6)$$

このとき、 $I \in \mathbb{R}^{m \times m}$ は単位行列、 $\gamma > 0$ は与えられた定数を表す。式 (6) は X と W の両凸 2 次最小化問題であることから、この問題の解は X と W について交互に最適化を行うことにより取得出来る。 W の最適解は特異値分解を用いて解くことができるが、計算時間を削減

するため、文献 [1] と同様に勾配降下法を用いたアルゴリズムを使用する。

逐次解法における t 回目の更新則として、 $X^{(t-1)}$ を固定した時の W の勾配は、

$$\begin{aligned} \nabla_W^{(t)} &= \frac{1}{2} \frac{\partial}{\partial W} f_\gamma(W^{(t-1)}, X^{(t-1)}) \\ &= \gamma W^{(t-1)} + (W^{(t-1)} - I) X^{(t-1)} X^{(t-1)T} \end{aligned} \quad (7)$$

で表される。この時、 $f_\gamma(W, X)$ は

$$f_\gamma(W, X) = \|(I - W)X\|_F^2 + \gamma \|W\|_F^2 \quad (8)$$

で定義される。また、 $W^{(0)}$ はゼロ行列とする。同様に、 $W^{(t)}$ を固定した時の X の勾配は、

$$\begin{aligned} \nabla_X^{(t)} &= \frac{1}{2} \frac{\partial}{\partial X} f_\gamma(W^{(t)}, X^{(t-1)}) \\ &= (I - W^{(t)})^T (I - W^{(t)}) X^{(t-1)} \end{aligned} \quad (9)$$

で表される。このとき、 $X^{(0)}$ は $P_\Omega(X^*)$ とする。

式 (6) は、式 (7) と式 (9) で示される勾配と、 W と X のそれぞれに使用するステップサイズパラメータを表す固定値 μ, τ を用いて、以下の AGD に基づく手続きを繰り返すことによって近似解を得ることが出来る。

$$\begin{aligned} \text{Step1} & \quad \nabla_W^{(t)} \leftarrow (7) \\ \text{Step2} & \quad W^{(t)} \leftarrow W^{(t-1)} - \mu \nabla_W^{(t)} \\ \text{Step3} & \quad \nabla_X^{(t)} \leftarrow (9) \\ \text{Step4} & \quad X^{(t)} \leftarrow P_\Omega^c(X^{(t-1)} - \tau \nabla_X^{(t)}) + P_\Omega(X^{(0)}) \end{aligned} \quad (10)$$

4 章冒頭でも示される問題点として、AGD はステップサイズパラメータである μ, τ の値によって振る舞いが大きく変動し、適切な値を選ぶことが非常に困難であることが挙げられる。また適切な値でない場合、収束に時間がかかることがある。そこで本論文では AGD に対して深層展開を適用することで最適なステップサイズを導出し、高速化を図る手法を提案する。

まず、AGD の発展的解法である OAGD (Optimized AGD) を説明する。同手法は勾配 $\nabla_W^{(t)}, \nabla_X^{(t)}$ を用いて式 (6) の目的関数を最小化する際に、各繰り返しにおいて常にその目的関数を最小化させるステップサイズパラメータ $\mu^{(t)}, \tau^{(t)}$ を決定する手法である。 μ と τ の更新式は以下の様に定義される。

$$\mu^{(t)} = \frac{\|\nabla_W^{(t)}\|_F^2}{\|\nabla_W^{(t)} X^{(t-1)}\|_F^2 + \gamma \|\nabla_W^{(t)}\|_F^2} \quad (11)$$

$$\tau^{(t)} = \frac{\|\nabla_X^{(t)}\|_F^2}{\|(I - W) \nabla_X^{(t)}\|_F^2} \quad (12)$$

式 (11), (12) を用いて OAGD の手続きは以下の様に示される。

$$\begin{aligned}
\text{Step1} \quad & \nabla_W^{(t)} \leftarrow \frac{\partial}{\partial W} f_\gamma(W^{(t-1)}, X^{(t-1)}) \\
\text{Step2} \quad & \mu^{(t)} \leftarrow (11) \\
\text{Step3} \quad & W^{(t)} \leftarrow W^{(t-1)} - \mu^{(t)} \nabla_W^{(t)} \\
\text{Step4} \quad & \nabla_X^{(t)} \leftarrow \frac{\partial}{\partial X} f_\gamma(W^{(t)}, X^{(t-1)}) \quad (13) \\
\text{Step5} \quad & \tau^{(t)} \leftarrow (12) \\
\text{Step6} \quad & X^{(t)} \leftarrow P_\Omega^c(X^{(t-1)} - \tau^{(t)} \nabla_X^{(t)}) \\
& \quad \quad \quad + P_\Omega(X^{(0)})
\end{aligned}$$

次に、OAGD に深層展開を適用した提案手法である TAGD (Trainable AGD) について説明する。本手法は OAGD に基づき、 $\mu^{(t)}$ と $\tau^{(t)}$ を最適な値に調整するための学習可能パラメータ $\eta_\mu^{(t)}$, $\eta_\tau^{(t)}$ を用いて、 $\mu^{(t)}$ と $\tau^{(t)}$ をそれぞれ $\eta_\mu^{(t)} \mu^{(t)}$, $\eta_\tau^{(t)} \tau^{(t)}$ に置き換えることで OAGD の収束の高速化を図る。 $\eta_\mu^{(t)}$, $\eta_\tau^{(t)}$ は以下に示される最適化問題により学習される。

$$\text{Maximize}_{\{\eta_\mu^{(t)}, \eta_\tau^{(t)}\}_{t=1}^T} \sum_{(X^*, \Omega) \in D} \sum_{t=1}^T -10 \log_{10} \frac{\|X^* - X^{(t)}\|_F^2}{\|X^*\|_F^2} \quad (14)$$

このとき、 D はデータセット集合、 T は最大繰り返し回数を表す。 D における X^* と Ω は一様分布に基づく乱数により生成することで、式 (14) における真の行列と観測集合を事前に用意することが可能である。同問題は OAGD における $\mu^{(t)}$, $\tau^{(t)}$ を学習可能パラメータ $\eta_\mu^{(t)}$, $\eta_\tau^{(t)}$ を用いて調整し、それにより得られる出力値 $X^{(t)}$ と真の行列 X^* の誤差を最大化する問題である。

本研究では式 (14) の目的関数を最大化させるような $\eta_\mu^{(t)}$, $\eta_\tau^{(t)}$ を深層展開を用いて学習し、そのパラメータを用いて導出される最適なステップサイズを用いることで AGD の高速化を図る。なお、それぞれのパラメータ最適化には Adam を使用した。

3.2. 行列因子分解に基づく行列補完

行列因子分解を用いた行列補完では、行列 $X \in \mathbb{R}^{M \times N}$ ($M < N$) は 2 つの行列 $U \in \mathbb{R}^{k \times M}$ と $V \in \mathbb{R}^{k \times N}$ を用いて、 $X = U^T V$ の形で表されると仮定する。この仮定に基づき、行列補完問題は以下の式で定式化される。

$$\text{Min}_{(m,n) \in \Omega} \sum (x_{m,n}^* - U_{:,m}^T V_{:,n})^2 + \gamma (\|U\|_F^2 + \|V\|_F^2) \quad (15)$$

ここで、 Ω は観測可能な要素の添字集合を、 $x_{m,n}^*$ は観測可能な真の行列要素を、 $U_{:,m}$, $V_{:,n}$ はそれぞれ U , V の第 m , n 列ベクトルを、 $\gamma > 0$ は正則化係数を示す。同問題における AGD は U , V に関する目的関数の勾配を利用し、交互に勾配法を適用することを繰り返す解法だが、上述した NSAO 法に基づく行列補完手法と同様に、ステップサイズの値によって振る舞いが大きく変動し適切な値を選ぶことが非常に困難である。そこで、アルゴリズム内部に埋め込んだパラメータを学習させることによる深層展開を用いてこの問題を解決する。

それに先立ち、行列因子分解を用いた行列補完に関する AGD について、その発展的発展的解法である OAGD を説明する。同手法は勾配 $\nabla_U^{(t)}$, $\nabla_V^{(t)}$ を用いて式 (15) を

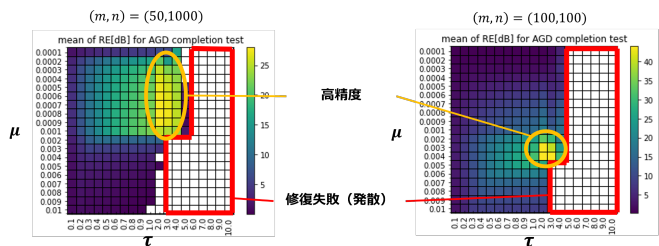


図 2. NSAO 法に対する事前実験の結果

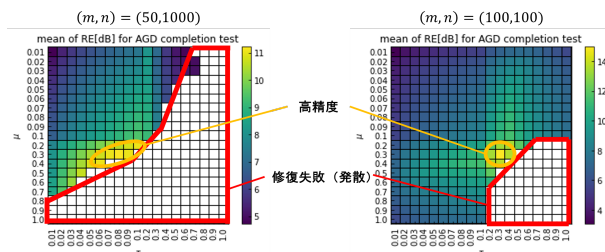


図 3. 行列因子分解に基づく行列補完手法に対する事前実験の結果

最小化する際に、各繰り返し時刻 t において常にその目的関数を最小化させるステップサイズパラメータ $\mu^{(t)}$, $\tau^{(t)}$ を決定する手法である。以下に OAGD の手続きを示す。

1. $\nabla_U^{(t+1)} \leftarrow \gamma U^{(t)} + V^{(t)} V^{(t)T} U^{(t)} - V^{(t)T} X^{(t)T}$
2. $\mu^{(t+1)} \leftarrow \|\nabla_U^{(t+1)}\|_F^2 / \left(\|\nabla_U^{(t+1)T} V^{(t)}\|_F^2 + \gamma \|\nabla_U^{(t+1)}\|_F^2 \right)$
3. $U^{(t+1)} \leftarrow U^{(t)} - \mu^{(t+1)} \nabla_U^{(t+1)}$
4. $\nabla_V^{(t+1)} \leftarrow \gamma V^{(t)} + U^{(t+1)} U^{(t+1)T} V^{(t)} - U^{(t+1)} X^{(t)}$
5. $\tau^{(t+1)} \leftarrow \|\nabla_V^{(t+1)}\|_F^2 / \left(\|\nabla_V^{(t+1)T} U^{(t+1)}\|_F^2 + \gamma \|\nabla_V^{(t+1)}\|_F^2 \right)$
6. $V^{(t+1)} \leftarrow V^{(t)} - \tau^{(t+1)} \nabla_V^{(t+1)}$
7. $x_{m,n}^{(t+1)} \leftarrow U_{:,m}^{(t+1)T} V_{:,n}^{(t+1)}$ for $(m, n) \notin \Omega$

(16)

同式において μ, τ は、勾配 ∇_U, ∇_V を用いて U, V を更新する際に式 (15) の目的関数を最小にするステップサイズとなる。

本節においても前節と同様に、 $\mu^{(t)}$ と $\tau^{(t)}$ を最適な値に調整するための学習可能パラメータ $\eta_\mu^{(t)}$, $\eta_\tau^{(t)}$ を用いて、 $\mu^{(t)}$ と $\tau^{(t)}$ をそれぞれ $\eta_\mu^{(t)} \mu^{(t)}$, $\eta_\tau^{(t)} \tau^{(t)}$ に置換することで OAGD の収束の高速化を図る。 $\eta_\mu^{(t)}$, $\eta_\tau^{(t)}$ は式 14 を最大化させるような $\eta_\mu^{(t)}$, $\eta_\tau^{(t)}$ を深層展開を用いて学習し、そのパラメータを用いて導出される最適なステップサイズを用いることで AGD の高速化を図る。こちらも、それぞれのパラメータ最適化には Adam を使用した。

4. 数値実験

本章では、行列補完問題に対して NSAO 法と行列因子分解による行列補完に使用される AGD, OAGD, 提案手法 (TAGD) を適用した結果を比較する。

事前実験として AGD における μ と τ の設定が困難であることを示す実験を行った。事前実験では各行列補完手法における固定値であるステップサイズパラメータ μ, τ の組み合わせにおいて、修復精度 (相対推定誤差の dB 値) を比較する実験を行った。 μ は [0.0001:0.01],

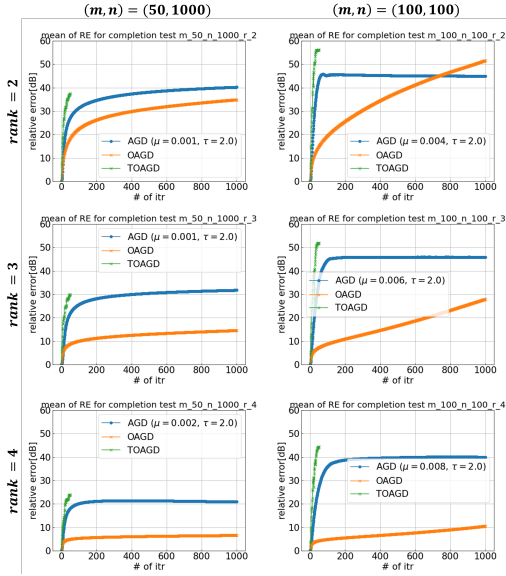


図 4. NSAO 法に基づく行列補完手法の実験結果

τ は $[0.1:10]$ の範囲で変動させており、修復対象である行列のサイズは正方行列とそうでない行列を想定した $(m, n) = (50, 1000), (100, 100)$ の 2 パターンで、ランクを 2 とするの行列を用いて行列補完を行った。図 2, 3 にそれぞれの行列のサイズに対する実験結果を示す。なお、評価は $RE = -10 \log_{10} \frac{\|X^* - X^{(T)}\|_F}{\|X^*\|_F}$ [dB] を推定精度の指標として用いている。図 2,3 における白で表される部分は、パラメータが大きすぎるために推定行列が発散することによって修復が失敗することを示している。結果を確認すると、いずれの結果もパラメータの設定による修復精度の増減が激しいことが分かる。また、修復精度が高い部分と修復が失敗する部分が近いことが確認できる。この結果から、AGD におけるステップサイズパラメータ μ と τ の設定が非常に困難であることが分かる。

続いて、AGD, OAGD, 提案手法 (TAGD) のそれぞれの条件における修復精度の比較実験結果を図 4, 5 に示す。本実験で用いた行列のサイズは $(m, n) = (50, 1000), (100, 100)$ の 2 パターンで、学習、テストに用いた行列のランクは $r = 2, 3, 4$ の 3 パターンを設定した。行列の欠損率は 80% とし、アルゴリズムの繰り返し回数は各種アルゴリズムは 1000 回、TAGD は 50 回としている。また、式 (14) で使用されている固定値 γ は修復精度にはほとんど影響しないため、 $\gamma = 10^{-1}$ で固定されている。提案手法において、学習にはエポック数が 1000 でミニバッチ数が 100 の合計 100000 要素のデータセットをランダムに生成して用いた。図 4, 5 に比較実験結果を示す。グラフの縦軸は RE を示しており、100 回の試行をして得られた平均を用いている。また、グラフの横軸は繰り返し回数を示している。なお AGD の結果は、事前実験の結果を得たときのように μ, τ を最適な値に設定したときのものを用いている。図 4, 5 を確認すると、検証した全てのパターンにおいて提案手法である TAGD の収束速度が AGD や OAGD に比べて大きく向上していることが分かる。図 4 の NSAO 法に基づく提案手法においては、繰り返し回数が 20 回あたりの時点で AGD の 50 回時点における推定精度に近い値を示しており、2 倍以上の収束速度を保っていることが確認できる。また、図 5 の行列因子分解に基づく手法においても、AGD や OAGD は 300~700 回程度で最終的な推定精度に達しているのに対し、提案手法は 50 回程度で達しており約 6 倍以上の収束速度を保っていることが

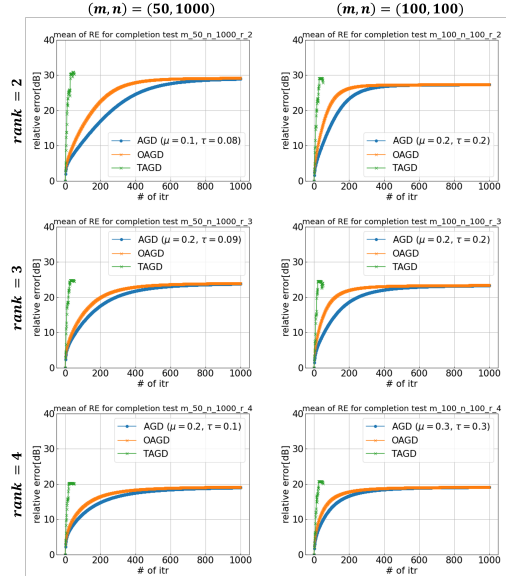


図 5. 行列因子分解に基づく行列補完手法の実験結果

確認出来る。どちらの行列補完手法においても最終的な推定精度が大きく向上しており、最良のパラメータを用いた AGD よりも約 10dB 以上推定精度が向上していることが確認できる。加えて、OAGD と比べても十分に優れた推定精度を持つことが確認できた。

5. おわりに

本論文では、交互勾配降下法を用いた行列補完である NSAO 法と行列因子分解に基づく行列補完手法に対して学習可能パラメータを埋め込み、そのパラメータを学習させる深層展開を適用することでステップサイズを最適化させる Trainable AGD を提案した。実験によってその有用性を検証し、既存のアルゴリズムの結果と比較して収束速度と修復精度が向上することを確認した。今後は学習可能パラメータのロバスト性の検証や他の既存アルゴリズムに対する深層展開の適用を検討する。

参考文献

- [1] K. Konishi: Parallel GPU implementation of null space based alternating optimization algorithm for large-scale matrix rank minimization; IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012
- [2] 佐々木亮平, 小西克巳, 古川利博: 多様体低次元化に基づく行列因子分解アルゴリズムの提案と協調フィルタリングへの応用; 電子情報通信学会論文誌 D, Vol. J103-D, No. 6, pp.506-517, Jun., 2020
- [3] R. Sasaki and K. Konishi: Acceleration Technique for Multiple-means Clustering based Locally Lowrank Approach to Nonlinear Matrix Completion; 29th European Signal Processing Conference (EU-SIPCO), 2021
- [4] 佐々木亮平, 小西克巳, 兩車和憲, 高橋智博, 古川利博: 行列ランク最小化とスパース正則化を用いた複数の部分空間同定に基づく行列完成手法の提案; 電子情報通信学会 32nd SIP シンポジウム, 2017
- [5] 和田山正, 高邊賢史: 深層展開に基づく信号処理アルゴリズムの設計—収束加速とその理論的解釈; 電子情報通信学会 基礎・境界サイエンス Fundamentals Review, Vol. 14, No. 1, pp. 60-72, 2020
- [6] 内藤凜, 佐々木亮平, 小西克巳: 深層展開に基づく行列完成手法の高速化; 第 66 回システム制御情報学会研究発表講演会, 2022
- [7] 内藤凜, 佐々木亮平, 小西克巳: 深層展開を用いた行列因子分解に基づく行列補完手法の高速化; 第 37 回信号処理シンポジウム, 2022