個人情報保護のためのネットワーク上での分散型自己符号化器の研究

代表研究者　　小西　克巳　　法政大学情報科学部　教授
共同研究者　　佐々木　亮平　　東京工科大学コンピュータサイエンス学部　講師

1 はじめに

本研究は、ネットワーク上で自己符号化器による機械学習を実現する分散自己符号化器手法を扱う。自己符号化器とは、オートエンコーダ（Autoencoder）とも呼ばれる教師なし学習手法の一つで、深層学習にも応用される手法ある。入力データを情報圧縮（または次元圧縮）し、圧縮されたデータを展開して出力データを得るとき、入力データと出力データが同じになるような情報圧縮（次元圧縮）のパラメータを求めることで、入力データの特徴を求める教師なし学習手法である。例えば量販店やコンビニエンスストアなどで、個人に紐付けされた売上データに対して同学習手法を用いることで、「どのような人が、どのような商品を買うのか」という特徴を抽出することが可能となる。このとき、より大規模なデータを利用する方が、より精度の高い結果を得ることが出来る。しかしながら、個人に紐づけられたデータを一つの計算サーバで大量に扱うことは、情報漏洩のリスクが高く、個人情報保護の観点から好ましいことではない。近年、このような情報漏洩リスクを避けるため、ネットワーク上での分散型機械学習手法の研究が進められている。地域や国ごとに個人データを分散させ、分散させたサーバ上で機械学習を行い、学習パラメータのみの同期通信により精度の高い結果を得る手法である。このような分散型機械学習手法の研究の多くは教師あり学習を対象としており、教師なし学習を対象とした分散アルゴリズムの研究はほとんどない。教師あり学習では学習パラメータが陽に存在するため、同パラメータを同期するアルゴリズムが構築しやすいのに対し、教師なし学習では学習パラメータが陽に存在しないことが多く、分散アルゴリズムの構築が難しいためである。そこで本研究では、教師なし学習手法の一つである自己符号化器の分散アルゴリズムの構築とその数理的性質を導出する。

本研究では、自己符号化器の一つである低ランク行列補完問題を扱い、分散アルゴリズムを導出する。低ランク行列補完問題とは、行列の成分のうち僅かな数の一部の要素のみが既知で他が未知の場合に、未知の成分を推定する問題である。画像修復や音声修復、協調フィルタリングなど、様々な応用問題に適用が可能な問題である。同問題は一般にはNP困難な問題である[1]が、同問題を解く手法として、行列の核ノルム最小化に基づく手法が提案されている[2]。その中でもSingular Value Thresholding（SVT）法[3]と呼ばれる特異値分解を用いた手法が最も利用されている。同手法はアルゴリズムの収束性が示され、さらに、核ノルム最小化問題のラグランジュ緩和問題の最適解を与えることが示されている。本研究ではSVT法に基づき、分散アルゴリズムを構築した。

分散アルゴリズム導出において重要な点は、非分散アルゴリズムと比較したときの解の精度である。本研究では解の精度を担保するため、分散アルゴリズムの数理的性質を導出する。また、数値例により、導出された分散アルゴリズムの有効性を示す。

2 数学的準備

2-1 低ランク行列補完問題

低ランク行列補完問題は以下のような問題である。

$$\text{Minimize rank}(X) \ \text{ subject to } X_{ij} = \bar{X}_{ij} \ \text{ for all } (i,j) \in \Omega$$

ただし、行列Xは設計変数、X_{ij}は行列Xの(i,j)成分、$\bar{X}$は与えられた行列、Ωは行列Xの要素のうち既知の成分の集合である。つまり、行列Xに未知の成分があるとき、行列が低ランクになるように未知の成分を補完する問題である。上記問題は行列ランク最小化問題と呼ばれ、一般にはNP困難であることが示されている。そこで、上記問題の近似解を与えるため、以下の行列の核ノルム最小化問題が提案されている。

$$\text{Minimize } \|X\|_* \ \text{ subject to } X_{ij} = \bar{X}_{ij} \ \text{ for all } (i,j) \in \Omega$$

ただし、$\|X\|_*$は行列Xの核ノルム、すなわち、特異値の和を表す。上記問題を解くことで、行列ランク最小

公益財団法人電気通信普及財団
研究調査助成報告書　第40号　2025年度

化問題の良い近似解が得られることが知られている。また、行列の核ノルム最小化問題は凸最適化問題であり、様々な解法が提案されている。内点法に基づくアルゴリズムにより厳密解を得ることができるが、計算メモリと計算時間を要することから、次に紹介する Singular Value Thresholding（SVT）法に基づくアルゴリズムが多く用いられている。

2-2 SVT 法

SVT 法は行列の特異値分解に基づくアルゴリズムである。アルゴリズムを以下の Algorithm 1 に示す。

Algorithm 1 Fixed point iterative algorithm.

Require: $X^0, \nu, \bar{X}, \Omega$

1: $k \leftarrow 0$
2: **repeat**
3: $\quad Y^{k+1} \leftarrow \mathcal{P}_\Omega(X^k)$
4: $\quad X^{k+1} \leftarrow S_\nu(Y^{k+1})$
5: $\quad k \leftarrow k+1$
6: **until** converge

Ensure: X^k

同アルゴリズムにおいて、P_Ωは$(i,j) \in \Omega$である(i,j)成分のX_{ij}に$\bar{X}_{ij}$を代入する演算子であり、S_νは行列Xの全ての特異値をνだけ小さくし、0 より小さくなる場合は、その特異値の値を 0 とする演算子である。つまり、上記アルゴリズムでは、特異値を小さくすることと、既知の値を代入することを収束するまで繰り返すことで、低ランク行列を求めている。SVT 法は収束することが証明されており、以下の問題の厳密解を与えることが示されている。

$$\text{Minimize } \nu\|X\|_* + \frac{1}{2}\sum_{(i,j)\in\Omega}\left\|X_{ij} - \bar{X}_{ij}\right\|_F^2$$

上記問題は、前述の行列の核ノルム最小化問題のラグランジュ緩和問題であり、Algorithm 1 は行列の核ノルム最小化問題の良い近似解を与える。同アルゴリズムは特異値分解に基づくアルゴリズムであるため、GPU 計算により高速化することが可能という特徴を持つ。多くの数値計算により同アルゴリズムの有効性が示されている。

3 分散アルゴリズム

本節では、前節で紹介した Algorithm 1 に基づいて分散アルゴリズムを導出する。行列Xを以下のようにL個に分割し、

$$X = [X_1\ X_2 \dots X_L]$$

以下のような行列X_iに関する核ノルム最小化問題を考える。

$$\text{Minimize } \nu\|X_i\|_* + \frac{1}{2}\sum_{(i,j)\in\Omega}\left\|X_{ij} - \bar{X}_{ij}\right\|_F^2$$

上記問題は分割された行列X_iに対して Algorithm 1 を適用することで解くことができる。しかしながら、行列を分割することでヒントとなる既知成分の情報が減り、分割すればするほど精度は劣化する。

行列Xがランク rであることを仮定し、行列X_iと行列X_jの左特異値ベクトル行列U_iとU_jを以下のように表現すると

$$U_i = [u_1^i \dots u_r^i\ u_{r+1}^i \dots u_m^i]$$

$$U_j = [u_1^j \dots u_r^j\ u_{r+1}^j \dots u_m^j]$$

ベクトル $u_1^i \dots u_r^i$ が貼る空間とベクトル $u_1^j \dots u_r^j$ が貼る空間は同じであることから、

$$u_p^i = \sum_{q=1}^{r} \left({u_p^i}^T u_q^j \right) u_q^i$$

が成り立つ。これを行列で表現すると

$$U_i^r = U_j^r {U_j^r}^T U_i^r$$

が成り立つ。ただし、

$$U_i^r = [u_1^i \ \ldots \ u_r^i \ ]$$

$$U_j^r = [u_1^j \ \ldots u_r^j]$$

である。両編を jについて 1 から rまで足して L で割ると

$$U_i^r = \frac{1}{L} \sum_{j=1}^{L} U_j^r {U_j^r}^T U_i^r$$

が成り立つ。この事実に基づき、行列X_iに Algorithm 1 を適用しつつ、同期を実行する Algorithm 2 を提案する。

Algorithm 2 Distributed matrix shrinkage iterative algorithm.

Require: $\nu, r, W^0, X^0, \Omega_i$ for $i = 1, 2, \ldots, L$

1: $k \leftarrow 0$
2: $[X_1^0 \ X_2^0 \ \ldots \ X_L^0] \leftarrow X^0$
3: **repeat**
4: **for** $i = 1$ to L **do**
5: $Y_i^{k+1} \leftarrow \mathcal{F}_{W^k}(X_i^k)$
6: $Z_i^{k+1} \leftarrow \mathcal{P}_{\Omega_i}(Y_i^{k+1})$
7: $X_i^{k+1} \leftarrow \mathcal{S}_\nu(Z_i^{k+1})$
8: $U_i^{k+1} \leftarrow \mathcal{U}_r(X_i^{k+1})$
9: **end for**
10: $W^{k+1} \leftarrow \frac{1}{L} \sum_{j=1}^{L} U_j^{k+1} {U_j^{k+1}}^T$
11: $k \leftarrow k + 1$
12: **until** converge

Ensure: $[X_1^k \ X_2^k \ \ldots \ X_L^k]$

Algorithm 2 においてF_Wは行列Wによる射影を表し、U_rは特異値の大きい順で r 番目の特異値に対応する左特異ベクトルを並べた行列を表す。３行から９行までが分割された行列に対する Algorithm 1 であり、10 行目で同期を実行している。このとき、以下の定理が成り立つ。

定理１

Algorithm 2 は収束する。

定理２

Algorithm 2 は以下の核ノルム最小化問題の厳密解を与える。

$$\text{Minimize } \nu \|X_i\|_* + \frac{1}{2} \sum_{(i,j) \in \Omega} \left\| F_W(X_{ij} - \bar{X}_{ij}) \right\|_F^2$$

それぞれの定理の証明は省略する。定理２はΩが十分な添字を含んでいれば、Algorithm 1 と Algorithm 2 が同じ解を与えることを意味している。分割した場合でも精度の高い解が得られること示唆しており、次節の数値実験により手法の有効性を示す。

4 数値実験

提案手法の有効性を示すため数値実験を行なった。全ての計算は、NVIDIA GeForce TRX 3090（メモリ 24GB）の GPU 上で実施し、MATLAB 2022b の Parallel Computing Toolbox を利用した。

最初に、分散アルゴリズムの有効性を示すため、Algorithm 1 と Algorithm 2 の比較を行なった。$4{,}000 \times 1{,}000$ の大きさのランク 3 の行列に対し、成分の 10%が既知であるとして 90%の未知の成分を推定する実験を行なった。分割数である L は 1（分割なし）と 4 とした。L=4 の場合に、同期ありと同期なしの場合を実行し比較した。その結果を図 1 に示す。縦軸が真値との相対誤差であり、横軸は繰り返し数である。

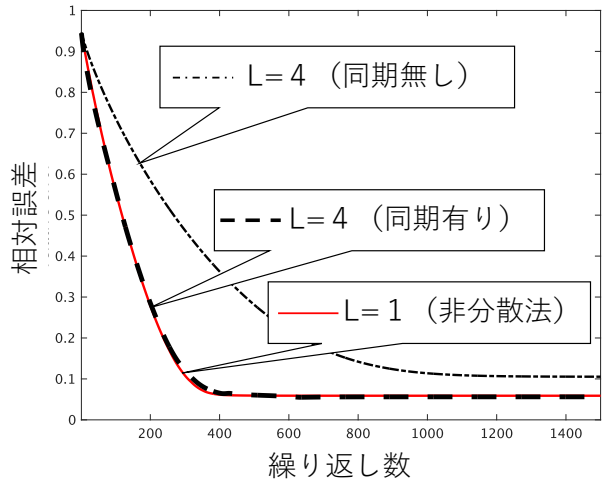

図 1　比較結果

L= 1 の場合と L=4（同期あり）がほとんど同じ精度であることが確認できる。同期がない場合は、解の収束速度と精度が劣化していることが分かる。以上から、分散アルゴリズムである Algorithm 2 が有効であることが確認された。

次に超大規模行列に Algorithm 2 を適用した。$2{,}000 \times 1{,}000{,}000$の大きさのランク 10 の行列に対して Algorithm 2 を L=2, 4, 8, 16 の場合に適用した。その結果を表 1 に示す。分割数が増えるほど解の精度が劣化しているのが、極端に精度が劣化していないことが確認できる。

表 1　超大規模行列への適用結果

L	相対誤差
2	4.755×10^{-4}
4	8.492×10^{-4}
8	1.080×10^{-3}
16	1.568×10^{-3}

5 まとめ

本研究では、自己符号化器の一つである低ランク行列補完問題を扱い、分散アルゴリズムを導出した。これにより、ネットワーク上で自己符号化器による機械学習を実現する分散自己符号化器を実現した。提案アルゴリズムは、対象となる行列の左特異ベクトルを同期することにより、行列の列ベクトルが貼る空間を同期することで解の精度を担保する手法である。数値実験により、提案する分散アルゴリズムが、非分散アルゴリズムと同等の性能を有することが確認された。分割数を増やすと解の精度は劣化するが、分割数などに対してどの程度劣化するかを数理的に明らかにすることが今後の課題である。

本研究では低ランク行列補完問題のみを扱ったが、自己符号化器の本質は次元削減であり、その本質は同じである。提案手法は主成分分析における主成分を同期する方法であり、自己符号化器は主成分分析の拡張

ととらえることができる。すでに我々は、非線形な低ランク行列補完問題、すなわち、行列の列または行ベクトルが多様体上に存在する場合の行列補完問題の解法を提案しており、これらの方法と本研究の分散アルゴリズムを融合し拡張することで、一般的な分散型自己符号化器の実現が可能と考えられる。このような手法を確立することで、個人情報を保護するネットワーク上での分散方自己符号化器の手法を構築することが、今後の課題である。

【参考文献】

[1] M. Fazel, “Matrix Rank Minimization with Applications,” Ph.D. dissertation, Stanford University, 2002.

[2] E. J. Candes and B. Recht, “Exact matrix completion via convex optimization,” Foundations of Computational Mathematics, vol. 9, no. 6, pp. 717–772, 2009.

[3] S. Ma, D. Goldfarb, L. Chen, “Fixed point and Bregman iterative methods for matrix rank minimization,” Mathematical Programming, vol. 128 (1), pp. 321–254, 2011.

〈発　表　資　料〉

題　　名	掲載誌・学会名等	発表年月
Adaptive Subspace Clustering for Matrix Completion	the 17th Asia Pacific Signal and Information Processing Association Annual Summit and Conference	2024 年 12 月
分散核ノルム最小化による低ランクテンソル補完手法	第 12 回計測自動制御学会制御部門マルチシンポジウム	2025 年 3 月