

超高次元データ解析のための 量子インスパイア主成分分析の開発

間島 慶[†] 小出 (間島) 真子[‡] 田桑 弘之[†]

樋口 真人[†] 須原 哲也[†] 八幡 憲明[†]

[†] 量子科学技術研究開発機構 〒263-8555 千葉県千葉市稲毛区穴川 4-9-1

[‡] 情報通信研究機構 〒184-8795 東京都小金井市貫井北町 4-2-1

E-mail: majima.kei@qst.go.jp

あらまし: 主成分分析は多変量データから重要な低次元成分を抽出する統計手法である。しかし、主成分分析のアルゴリズムは特異値分解に基づくため、次元数（変数の数）が数百万を超えるデータには、計算時間の問題からしばしば適用が困難となる。我々は近年発案された「量子インスパイアアルゴリズム」を用い、計算時間を次元数の対数オーダーに抑えつつ、主成分分析を近似するアルゴリズムを計算機実装した。本報告において、複数の人工データ・実データを用いてその計算時間と性能を評価した結果を紹介する。

キーワード: 主成分分析, 量子インスパイアアルゴリズム, 高次元データ

1. はじめに

主成分分析は多変量データから重要な低次元成分を抽出する統計手法である。与えられた多変量データを線形変換し、データ内の分散を最も保つ低次元成分を抽出する。多変量データの解析において頻繁に用いられる統計手法の一つであり、例えば神経科学の研究では多次元・多地点から計測された神経活動のデータから解釈を得るために用いられている[1], [2].

しかし、主成分分析は計算時間の問題から、高次元なデータへの適用はしばしば困難となる。特異値分解に基づいたアルゴリズムを用いた場合、取り扱うデータの変数の数（次元数）に対し、2乗で計算時間が増加する。近年ではデータの計測技術の進展・高解像度化により、次元数は数千万以上に達することがあり、汎用型 CPU 搭載の PC を用いた場合、数週間以上の計算時間を要する。

このような計算時間の増加に対処するため、本報告では量子インスパイアアルゴリズムを導入する。量子インスパイアアルゴリズムは量子機械学習アルゴリズムの有用性を検証する過程で提案された古典アルゴリズムであり、近似にはなるが、特異値分解の計算量を次元数の対数オーダーに抑えることができる[3]. この量子インスパイアアルゴリズムを用いることで線形回帰[4], [5], 主成分分析[6], 正準相関分析[7], 非負値行列分解[8], サポートベクトルマシン[9]などを高速に近似するアルゴリズムが提案されている。しかし、近似精度に関する理論的な研究がなされる一方で、それら量子インスパイアアルゴリズムは実際のデータ解析においてまだほとんど用いられておらず、その有用性は

未知数である。

本報告では、数百万～数億次元に達する高次元データ解析に向けて、量子インスパイアアルゴリズムによる主成分分析の計算時間と性能を計算機実験によって評価する。

2. アルゴリズム

この節では本報告の実験で用いるアルゴリズムについて説明する。主成分分析 (2.1 節), 量子インスパイアアルゴリズムを用いた主成分分析 (2.2 節), 量子インスパイアアルゴリズムを用いるためのデータ構造 (2.3 節) の順に説明を行う。

2.1. 主成分分析

サンプル数 N , 次元数（変数の数） D のデータ行列 $\mathbf{X} \in \mathbb{R}^{N \times D}$ が与えられたとする。データのセンタリング（平均を $\mathbf{0}$ にシフトする前処理）はすでになされているものとする。主成分分析では、第一主成分を抽出する重みベクトルとして、以下の条件を満たすベクトル $\hat{\mathbf{w}} \in \mathbb{R}^D$ を計算する。

$$\hat{\mathbf{w}} = \underset{\|\mathbf{w}\|=1}{\operatorname{argmax}} \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} . \quad (1)$$

ここで、 $\|\mathbf{w}\|$ はベクトル \mathbf{w} の L2 ノルム（ユークリッドノルム）を、 $\mathbf{X}^T, \mathbf{w}^T$ は行列 \mathbf{X} , ベクトル \mathbf{w} を転置したものを表す。第二主成分以降の重みベクトルは、上と同型の最適化問題をすでに得られている重みベクトルと直交する条件のもと解くことで得られる。データ解析では通常、上位の主成分のみに興味があることが多い。ここでは上位 K 個の主成分に興味があるとし、それに対する重みベクトルを $\hat{\mathbf{w}}^{(1)}, \hat{\mathbf{w}}^{(2)}, \dots, \hat{\mathbf{w}}^{(K)}$ と

記す。重みベクトルを計算するアルゴリズムは複数あるが、ここでは特異値分解を用いる方法を説明する。データ行列 \mathbf{X} の特異値分解を $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ とおく。第 k 番目の右特異ベクトルは第 k 主成分の重みベクトルと一致することが知られている。つまり、特異値分解の結果を用いて、 $\hat{\mathbf{w}}^{(k)} = \mathbf{V}(:,k)$ として重みベクトルを求めることができる。サイズ $N \times D$ の行列の特異値分解にかかる計算量は $O(\min(N^2D, ND^2))$ であるため、サンプル数 N が十分大きい時、その計算時間は次元数 D の 2 乗に比例する。

2.2. 量子インスパイア主成分分析

本報告の実験では先行研究 Koide-Majima & Majima [7] で使われている量子インスパイアアルゴリズムと同一のものを用いて主成分分析を行った。これは Tang [3] によって提案された特異値分解を行う量子インスパイアアルゴリズム（量子インスパイア特異値分解）をベースに作られたものである。本報告では 2.1 節で述べた主成分分析のアルゴリズムにおいて、特異値分解を量子インスパイア特異値分解に置き換えたものを量子インスパイア主成分分析と呼ぶ。

量子インスパイア特異値分解の概要を示す。前節と同様、データ行列 $\mathbf{X} \in \mathbb{R}^{N \times D}$ が与えられているとし、その特異値分解を $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ とする。量子インスパイア特異値分解では、右特異ベクトル $\mathbf{V}(:,k)$ を近似する description を計算する。description とは行列 $\mathbf{S} \in \mathbb{R}^{P \times D}$ とベクトル $\mathbf{u}_k \in \mathbb{R}^P$ ($k = 1, \dots, K$) の組で、 $\mathbf{V}(:,k)$ を $\mathbf{S}^T \mathbf{u}_k$ によって近似するものをさす。ここで P は近似精度と計算時間のトレードオフを制御するパラメータである。本報告では全て先行研究と合わせ $P = 150$ とした。量子インスパイア特異値分解では、2.3 節で説明するデータ構造を用いることで右特異ベクトルの description を $O(\log(ND))$ の計算量で得ることができる。以下、量子インスパイア特異値分解によって右特異ベクトルの description を計算する手順を説明する。

量子インスパイア特異値分解では、与えられたデータ行列から P 個の行と列を選び出し、サイズ $P \times P$ の行列を作成する。その行列に対し通常の特異値分解を行い、その結果を用いて元のデータ行列の右特異ベクトルを近似する description を構成する。サイズ $P \times P$ の行列を構成する際、その行・列は以下のルールで選ばれる。まず、行を選択するために、行番号の添字 $i \in \{1, \dots, N\}$ を以下の確率で返す離散確率分布から P 回サンプリングする：

$$\mathcal{F}(i) = \frac{\|\mathbf{X}(i,:)\|^2}{\|\mathbf{X}\|_F^2}. \quad (2)$$

ここで、 $\|\mathbf{X}\|_F$ は行列 \mathbf{X} のフロベニウスノルムを表す。サンプリングされた P 個の添字を i_1, i_2, \dots, i_P と

おく。このサンプリングは 2.3 節で述べる二分木を用いたデータ構造を用いることで、 $O(\log(N))$ オーダーの計算量で実行することができる。

行番号と同様に、列番号の添字 $j \in \{1, \dots, D\}$ も以下の離散確率分布から P 回サンプリングする：

$$\mathcal{G}(j) = \frac{1}{P} \sum_{p=1}^P \frac{\mathbf{X}(i_p, j)^2}{\|\mathbf{X}(i_p, :)\|^2}. \quad (3)$$

選ばれた添字を j_1, j_2, \dots, j_P とおく。これらの添字を用い、サイズ $P \times P$ の行列 \mathbf{W} を以下として構成する：

$$\mathbf{W}(p, q) = \frac{\mathbf{X}(i_p, j_q)}{P \sqrt{\mathcal{F}(i_p) \mathcal{G}(j_q)}}. \quad (4)$$

上記の手続きを擬似コードとしてまとめると以下となる。

アルゴリズム 1: Matrix Sampling
Input: データ行列 $\mathbf{X} \in \mathbb{R}^{N \times D}$, パラメータ P
Output: 行列 $\mathbf{W} \in \mathbb{R}^{P \times P}$, 添字 $\{i_p\}_{p=1}^P$

- 1: For $p = 1$ to P do
- 2: 確率分布 \mathcal{F} からサンプリングを行い、結果得られた添字を i_p とおく
- 3: End for
- 4: For $p = 1$ to P do
- 5: 確率分布 \mathcal{G} からサンプリングを行い、結果得られた添字を j_p とおく
- 6: End for
- 7: (p, q) 番目の要素が $\frac{\mathbf{X}(i_p, j_q)}{P \sqrt{\mathcal{F}(i_p) \mathcal{G}(j_q)}}$ となるサイズ $P \times P$ の行列 \mathbf{W} を定義する

量子インスパイア特異値分解では、上記で構成された行列 \mathbf{W} に通常の特異値分解を行う。その結果を $\mathbf{W} = \mathbf{U}_W \mathbf{\Sigma}_W \mathbf{V}_W^T$ とおく。得られた結果を用い、description を構成する行列 $\mathbf{S} \in \mathbb{R}^{P \times J}$ とベクトル $\mathbf{u}_1, \dots, \mathbf{u}_K \in \mathbb{R}^P$ を以下として定義する。 \mathbf{S} に関しては、 p 番目の行が $\mathbf{X}(i_p, :)$ と一致する行列として定義する。ベクトル \mathbf{u}_k に関しては、 p 番目の要素が

$$\frac{\mathbf{U}_W(p, k)}{\mathbf{\Sigma}_W(k, k) \sqrt{PF(i_p)}} \quad (5)$$

となるように定義する。上記のように \mathbf{S} , \mathbf{u}_k を定義すると、線形代数演算に関する乱択アルゴリズムの理論から、 $\mathbf{S}^T \mathbf{u}_k$ がデータ行列 \mathbf{X} の右特異ベクトル $\mathbf{V}(:,k)$ を近似することが知られている [3], [10]。以上の手続きによる量子インスパイア特異値分解の擬似コー

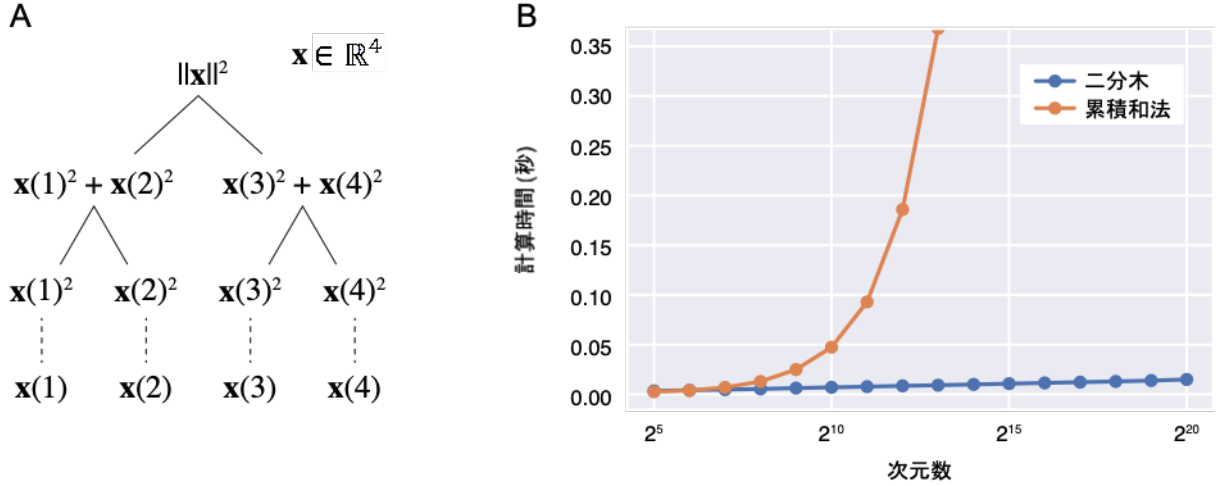


図 1. 二分木データ構造とそれを用いたサンプリング. (A) 二分木構造を用いたデータ構造. (B) サンプリングに必要な計算時間. 二分木を用いた場合と累積和法を用いた方法をそれぞれ 10 回ずつ行い, その計算時間の平均を次元数の関数としてプロットした.

ドを以下にまとめる.

アルゴリズム 2: quantum-inspired singular (qiSVD)

Input: データ行列 $\mathbf{X} \in \mathbb{R}^{N \times D}$, パラメータ K, P

Output: Description $\{\mathbf{S}, \{\mathbf{u}_k\}_{k=1}^K\}$, 添字 $\{i_p\}_{p=1}^P$

- 1: $[\mathbf{W}, \{i_p\}_{p=1}^P] \leftarrow \text{MatrixSampling}(\mathbf{X}, P)$
- 2: $[\mathbf{U}_W, \mathbf{\Sigma}_W, \mathbf{V}_W] \leftarrow \text{SVD}(\mathbf{W})$
- 3: p 行目を $\mathbf{X}(i_p, :)$ とする行列 $\mathbf{S} \in \mathbb{R}^{P \times J}$ を定義
- 4: p 番目の要素を $\mathbf{U}_W(p, k) / (\mathbf{\Sigma}_W(k, k) \sqrt{P \mathcal{F}(i_p)})$ とするベクトル $\hat{\mathbf{u}}_k \in \mathbb{R}^P$ ($k = 1, \dots, K$) を定義
- 5: グラム・シュミットの正規直交化法を $\{\mathbf{S}^T \hat{\mathbf{u}}_k\}_{k=1}^K$ に適用し, $\{\mathbf{S}^T \mathbf{u}_k\}_{k=1}^K$ が正規直交系となるようにベクトルの組 $\{\mathbf{u}_k\}_{k=1}^K$ を得る (詳細は[7]参照)

2.3. 二分木を用いたデータ構造

本節ではデータ行列 $\mathbf{X} \in \mathbb{R}^{N \times D}$ に対する量子インスパイア特異値分解を $O(\log(ND))$ で実行するためのデータ構造を説明する. 本報告で検証する量子インスパイア主成分分析は量子インスパイア特異値分解のアルゴリズムに基づいている. それ以外のステップは入力データの次元数 D に依存しない定数の計算量で実行できるため, 量子インスパイア主成分分析の次元数に対する計算量はそれぞれ $O(\log(D))$ となる. 以下, 量子インスパイア特異値分解を $O(\log(ND))$ の計算量で実行するためのデータ構造の説明を行う.

量子インスパイアアルゴリズムでは $P \times P$ の行列を構成する際, 確率分布 $\mathcal{F}(i)$, $\mathcal{G}(j)$ からサンプリングを行う (2.2 節参照). P を固定した場合, それ以外のステップの計算量は定数であり, 残るステップで

あるサンプリングを計算量を抑えて行うことが望まれる. 説明の単純化のため, ここではベクトル $\mathbf{x} \in \mathbb{R}^N$ が与えられ, その添字 n を確率分布 $P(n) = \mathbf{x}(n)^2 / \|\mathbf{x}\|^2$ からサンプリングすることを考える. 量子インスパイアアルゴリズムでは, ベクトルを図 1 A に示す二分木に格納することで, これを実行する. この二分木は末端にベクトルの各要素 (とその絶対値を 2 乗したもの) を格納し, 中間のノードにはそのノードの下部ノードのもつ値の和が格納されている. 二分木探索のアルゴリズムを用いることで, 与えられた値 $u \in [0, 1]$ に対し, 以下の条件を満たす添字 n' を $O(\log(N))$ の計算量で見つけ出すことができる:

$$\frac{1}{\|\mathbf{x}\|^2} \sum_{n=1}^{n'-1} \mathbf{x}(n)^2 \leq u < \frac{1}{\|\mathbf{x}\|^2} \sum_{n=1}^{n'} \mathbf{x}(n)^2. \quad (6)$$

そのため, 値 u を一様乱数からサンプリングすることによって, 上記のサンプリングを $O(\log(N))$ の計算量で実行できる. これは通常の線形時間を要するアルゴリズム (例えば Python の標準的なパッケージである NumPy に搭載されているサンプリングアルゴリズム) に比べ, はるかに高速にサンプリングを行うことができる (図 1B). このアルゴリズムを直接用いることで, 量子インスパイア特異値分解に含まれる確率分布 $\mathcal{F}(i)$ からのサンプリングを $O(\log(N))$ の計算量で行うことができる. また, データ行列の各列があらかじめ二分木で格納されているとすれば, 確率分布 $\mathcal{G}(j)$ からのサンプリングを $O(\log(D))$ の計算量で行うことができる. 注意点として, この二分木のデータ構造を用意するためには線形オーダーの計算量を要する. 本報告の以降の実験ではこの二分木のデータ構造の用

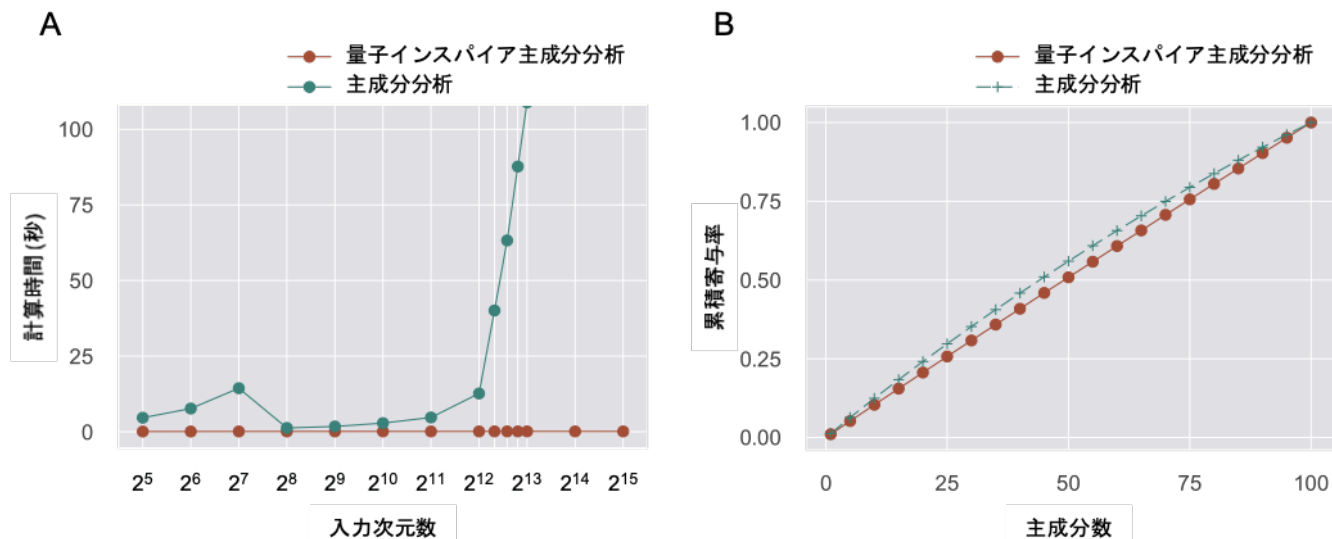


図 2. 主成分分析と量子インスパイア主成分分析のシミュレーションデータを用いた比較. (A) 計算時間の比較. 入力次元数を変えつつ, 10 回の平均計算時間をプロットした. (B) 累積寄与率による性能比較. 入力次元数を 10000 に固定し, 主成分数の関数として累積寄与率をプロットした.

意にかかる時間も含めて計算時間として評価している.

3. 実験設定

この節では実験で用いたデータ (シミュレーションデータと実ベンチマークデータセット), アルゴリズムの評価方法を説明する. いずれの実験も Intel CPU Xeon Gold 5115 (2.4 GHz, 768 GB memory) を用いて行った.

3.1. シミュレーションデータ

主成分分析, 量子インスパイア主成分分析の評価に以下の手続きで生成したシミュレーションデータを用いた. まず, 各要素が標準正規分布からサンプリングされた行列 $\mathbf{Z} \in \mathbb{R}^{N \times 100}$, $\mathbf{B} \in \mathbb{R}^{100 \times D}$ を作る. そして, サイズ $N \times D$ の行列 $\mathbf{X} = \mathbf{Z}\mathbf{B}$ を作り, それをデータ行列として用いた. 本報告の実験では $N = 10000$ とし, データの次元数 D は $\{2^5, 2^6, \dots, 2^{15}\}$ で変化させた.

3.2. 実ベンチマークデータ

本報告では実データとして主成分分析の評価に標準的に用いられる 5 つのデータセットを用いた. データの前処理は全て Koide-Majima & Majima [7] と同様の手続きで行った. ここではその概要のみを説明する.

最初の二つのデータセットは画像分野のものから選択した. データセット 1 は MNIST [11], データセット 2 は CIFAR10 [12] である. 本報告の実験ではピクセル値を特徴量とみなし, 主成分分析と量子インスパイア主成分分析を適用した.

3 つ目, 4 つ目のデータセットとしては, 言語分野のデータを用いた. データセット 3 は英語-ドイツ語の対訳文のデータセット WikiCLIR [13], データセット 4 は英語-日本語の対訳文のデータセット JESC [14] である. 各データセット内の英文を wikipedia2vec [15] という手法により 300 次元のベクトルに変換し, そのベクトルに対し, 主成分分析と量子インスパイア主成分分析を適用した.

5 つ目のデータセットとしては, 音声分野のデータセット XRMB [16] を用いた. XRMB は発話時の音声と喉の筋電応答の対データを提供しているデータベースである. 音声特徴量ベクトルに対し, 主成分分析, 量子インスパイア主成分分析を適用した.

3.3. シミュレーションデータ

神経活動のデータを用いて量子インスパイア主成分分析の計算時間・性能を検証するため, 画素数 621 万の二光子カルシウムイメージングデータを用いた. 二光子カルシウムイメージングとは, カルシウムイオンの濃度を可視化することにより, 神経細胞 (ニューロン) の活動を画像として計測する方法である. ニューロンは発火する際, 細胞体内のカルシウムイオンの濃度が上昇する. そのため, 二光子カルシウムイメージングでは, ニューロンの発火によって引き起こされるカルシウムイオンの濃度変化を捉えることができる. 本報告で用いるデータとして, マウスの脳から 1Hz で計測を行った. 画素数は 2500×2484 であり, マウスの非麻酔下の安静時脳活動を計測した. 画素値を特徴量

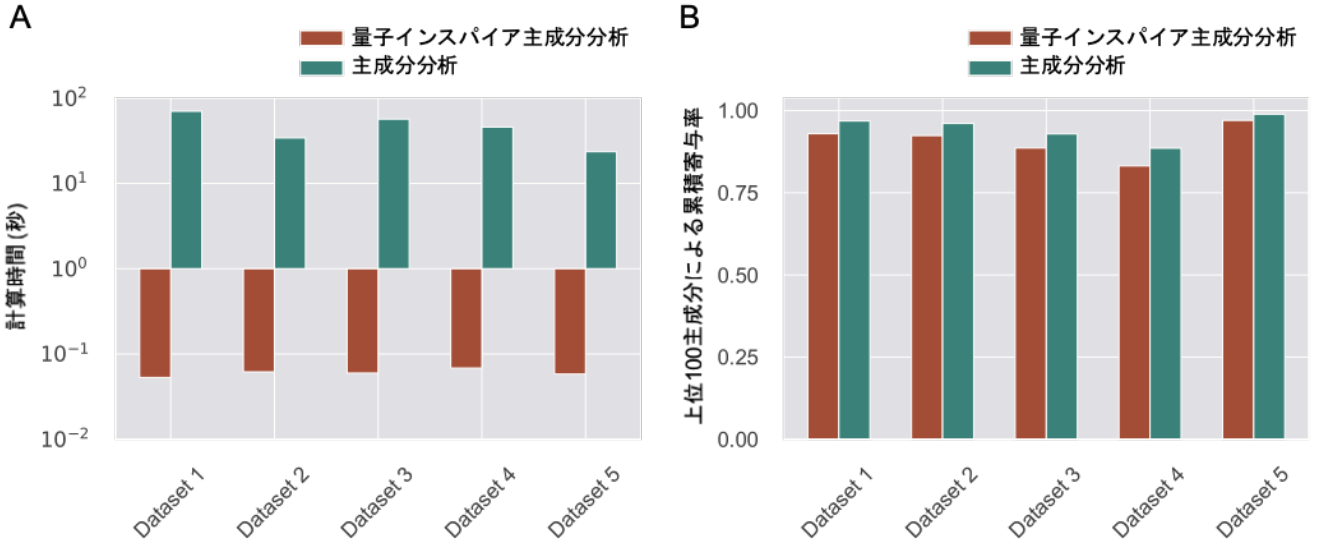


図 3. 主成分分析と量子インスパイア主成分分析の実データを用いた比較. (A) 計算時間の比較. 10 回の平均計算時間をプロットした. (B) 累積寄与率による性能比較. 上位 100 主成分を抽出し, その累積寄与率を評価した.

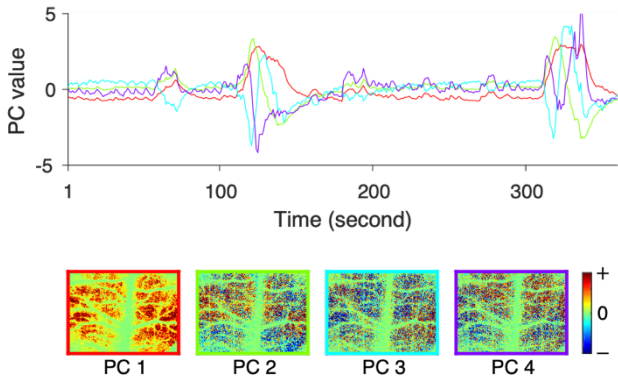


図 4. 高次元神経活動データへの量子インスパイア主成分分析適用. 画素数 621 万のマウス二光子カルシウムイメージングデータに量子インスパイア主成分分析を適用した. 得られた主成分のタイムコース (上段) と重みベクトル (下段).

とみなし, 621 万次元のベクトルに対し, 主成分分析と量子インスパイア主成分分析を適用した.

3.4. 評価指標

主成分分析, 量子インスパイア主成分分析の目的は与えられたデータ行列に含まれる変動を最も説明する低次元成分を抽出することである. そのため, 本報告では抽出された上位 100 個の低次元成分によって, 元のデータ行列 $\mathbf{X} \in \mathbb{R}^{N \times D}$ の値が何%復元できるかを以下の式で評価した:

$$1 - \frac{\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2}{\|\mathbf{X}\|_F^2}. \quad (7)$$

ここで, $\mathbf{W} \in \mathbb{R}^{D \times K}$ は主成分分析, または, 量子インスパイア主成分分析で抽出された上位 K 個の重みベクトルからなる行列 $[\hat{\mathbf{w}}^{(1)}, \hat{\mathbf{w}}^{(2)}, \dots, \hat{\mathbf{w}}^{(K)}]$ である.

4. 実験結果

4.1. シミュレーションデータを用いた評価実験

最初に, シミュレーションデータを用い, 主成分分析と量子インスパイア主成分分析を比較した. 入力データの次元数を指数関数的に増やしつつ, 計算時間を評価した (図 2A). 主成分分析の計算時間が指数関数的に増加していくのに対し, 量子インスパイア主成分分析では計算時間を改善することに成功している.

同じデータを用い, 次に主成分分析としての性能を評価した. 具体的には抽出した上位 k の主成分によってデータの分散がどれほど説明できるか (累積寄与率) を評価した (図 2B). 本基準においては量子インスパイア主成分分析の主成分分析に対する性能低下は 5%以内であった.

4.2. 実ベンチマークデータを用いた評価実験

次に 5 つの実ベンチマークデータセットを用い, 計算時間と性能の評価を行った (図 3A,B). 主成分分析と比較し, 量子インスパイア主成分分析による計算時間の改善が認められ, また, 性能の低下は最大 7%であった.

4.3. 神経データ解析への応用

神経活動データに対する量子インスパイア主成分分析の応用可能性を示すため, マウスの脳活動データ

を用いた実験を行った。本報告の検証では、画素数 621 万のマウス二光子イメージングデータを用いた(3.3 節参照)。画素データに対し、量子インスパイア主成分分析を適応した結果を図 4 として示す。消費した計算時間は 1 時間未満であり、これに対し、通常の主成分分析は同計算機環境において 24 時間以下で計算を終了させることができなかった。

5. 考察

本報告では先行研究において提案・実装した量子インスパイア主成分分析の計算時間、従来法に比べた性能をシミュレーションデータ、実ベンチマークデータ、神経活動データを用いて評価した。結果、性能低下を伴うものの、数百万次元以上の高次元データを扱うのにも耐えうる程度計算時間を改善することができた。また、今回用いたシミュレーションデータ、実ベンチマークデータセットにおいては、性能の低下は 7%以下であった。これらの結果は量子インスパイアアルゴリズムに基づく機械学習手法によって、これまで取り扱い自体が不可能であった大規模な高次元データを解析対象にできる可能性を示唆している。

謝辞

本研究は JSPS 科研費 JP20K16465 の助成を受けたものです。

文 献

- [1] J. P. Cunningham and B. M. Yu, “Dimensionality reduction for large-scale neural recordings,” *Nat Neurosci*, vol. 17, no. 11, pp. 1500–1509, Nov. 2014.
- [2] L. H. Nguyen and S. Holmes, “Ten quick tips for effective dimensionality reduction,” *PLoS Comput Biol*, vol. 15, no. 6, p. e1006907, Jun. 2019.
- [3] E. Tang, “A quantum-inspired classical algorithm for recommendation systems,” *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing - STOC 2019*, pp. 217–228, 2019.
- [4] N.-H. Chia, H.-H. Lin, and C. Wang, “Quantum-inspired sublinear classical algorithms for solving low-rank linear systems,” *arXiv:1811.04852 [quant-ph]*, Nov. 2018.
- [5] A. Gilyén, S. Lloyd, and E. Tang, “Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension,” *arXiv:1811.04909 [quant-ph]*, Nov. 2018.
- [6] E. Tang, “Quantum-inspired classical algorithms for principal component analysis and supervised clustering,” *arXiv:1811.00414 [quant-ph]*, Oct. 2018, Accessed: Jun. 23, 2019.
- [7] N. Koide-Majima and K. Majima, “Quantum-inspired canonical correlation analysis for exponentially large dimensional data,” *Neural Netw*, vol. 135, pp. 55–67.
- [8] Z. Chen, Y. Li, X. Sun, P. Yuan, and J. Zhang, “A Quantum-inspired Classical Algorithm for Separable Non-negative Matrix Factorization,” *arXiv:1907.05568 [cs]*, Jul. 2019.
- [9] C. Ding, T.-Y. Bao, and H.-L. Huang, “Quantum-Inspired Support Vector Machine,” *arXiv:1906.08902 [quant-ph, stat]*, Jul. 2019.
- [10] A. Frieze, R. Kannan, and S. Vempala, “Fast Monte-Carlo algorithms for finding low-rank approximations,” in *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, Nov. 1998, pp. 370–378.
- [11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [12] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” p. 60, 2009.
- [13] S. Schamoni, F. Hieber, A. Sokolov, and S. Riezler, “Learning Translational and Knowledge-based Similarities from Relevance Rankings for Cross-Language Retrieval,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland, 2014, pp. 488–494.
- [14] R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz, “JESC: Japanese-English Subtitle Corpus,” *arXiv:1710.10639 [cs]*, 2017.
- [15] I. Yamada *et al.*, “Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia,” *arXiv:1812.06280 [cs]*, 2018.
- [16] J. R. Westbury, “X-ray microbeam speech production database user’s handbook,” 1994. [Online]. Available: http://www.haskins.yale.edu/staff/gafos_downloads/ubdbman.pdf.