Vol.2021-QS-2 No.13
2021/3/29

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・正準相関分析の開発

間島慶[1,2]　小出（間島）真子[3,4]　八幡憲明[1]

概要：主成分分析・正準相関分析はともに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，これらのアルゴリズムは特異値分解に基づくため，次元数（変量の数）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・正準相関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．また，量子インスパイアアルゴリズムを用いた高速計算は単なる計算時間の削減にとどまらず，新しいデータ解析の方法を提供する．例として，与えられた多変量データ内の変量同士で積をとり，それを新しい変量とみなすことで次元数を増加させ，得られた高次元データに我々の開発した量子インスパイア正準相関分析を適用した．この操作はデータの次元数を増加させるため，通常の正準相関分析では計算時間の肥大化により取り扱いが困難となる．MNIST データセットを用いこれを行ったところ，提案法は線形のみの正準相関分析より多くの相関を抽出した．また抽出できた相関の量はカーネル正準相関分析，深層正準相関分析などの代表的な非線形手法と同程度であった．以上の結果は，量子インスパイアアルゴリズムが実データの解析において有用であり，従来では計算時間の問題から不可能であった超高次元データを扱う分野を開拓できる可能性を示している．

キーワード：主成分分析，正準相関分析，量子インスパイアアルゴリズム，高次元データ

Quantum-inspired principal component and canonical correlation analysis for high-dimensional data

KEI MAJIMA[†1,2]　NAOKO KOIDE-MAJIMA[3,4]
NORIAKI YAHATA[1]

Keywords: principal component analysis, canonical correlation analysis, quantum-inspired algorithm, high-dimensional data

1. はじめに *

主成分分析・正準相関分析はともに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．主成分分析は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]. 一方，正準相関分析は多変量データのペアからその二つのデータの間に共通する成分を線形変換により抽出する. 上記同様，神経科学研究の例として，ヒトの脳活動データと性格・精神疾患傾向・感情スコアのデータとの間の依存関係が正準相関分析によって明らかにされている [3]–[5]. このように主成分分析・正準相関分析は多変量データから有用な成分を抽出するツールとして，神経データ，画像，遺伝子データの解析などを中心に広く応用されている．

しかし，主成分分析・正準相関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．どちらのアルゴリズムも特異値分解に基づいているため，取り扱うデータの変量の数（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

* 1 量子科学技術研究開発機構
National Institutes for Quantum and Radiological Science and Technology
2 京都大学
Kyoto University
3 大阪大学
Osaka University
4 情報通信研究機構
National Institute of Information and Communications Technology

に抑えることができる [6]．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 [7], [8]，主成分分析 [9]，正準相関分析 [10]，非負値行列分解 [11]，サポートベクトルマシン[12]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，数千万〜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・正準相関分析の計算時間と性能を計算機実験によって評価する．

また，量子インスパイアアルゴリズムによる高速化は，単なる計算時間の短縮のみならず，新しい非線形な低次元成分・共通成分の抽出法を提供する．カーネル正準相関分析[13]–[16]や深層正準相関分析[17]では，カーネル法，深層ニューラルネットワークによる非線形変換を行い，その後正準相関分析を適用することで，線形変換のみでは見つけられない共通成分を抽出できる．本報告では，与えられたデータを 2 次の多項式により高次元空間に非線形変換し[18]，その後量子インスパイアアルゴリズムによる正準相関分析を適用した．この方法はデータの次元数を増加させるため，従来の正準相関分析では計算時間の問題から実行することが難しい．これによって，カーネル正準相関分析，深層正準相関分析と同様により良い共通成分を抽出できる例が確認された．

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），正準相関分析（2.3 節），量子インスパイアアルゴリズムを用いた正準相関分析（2.4 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（2.5 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（変量の数）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 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 $\widehat{\mathbf{w}} \in \mathbb{R}^D$ を計算する．

$$\widehat{\mathbf{w}} = \underset{\|\mathbf{w}\|=1}{\operatorname{argmax}}\ \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w}\ . \tag{1}$$

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，$\mathbf{X}^{\mathrm{T}}, \mathbf{w}^{\mathrm{T}}$ 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 $\widehat{\mathbf{w}}^{(1)}, \widehat{\mathbf{w}}^{(2)}, \cdots, \widehat{\mathbf{w}}^{(K)}$ 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 $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ 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，$\widehat{\mathbf{w}}^{(k)} = \mathbf{V}(:,k)$ 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 [10] 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 [6] 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 $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ 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 $\mathbf{u}_k \in \mathbb{R}^P$ $(k = 1, \cdots, K)$ の組で，$\mathbf{V}(:,k)$ を $\mathbf{S}^{\mathrm{T}}\mathbf{u}_k$ 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，2．5 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 $i \in \{1, \cdots, N\}$ 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：

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

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 $i_1, i_2, \cdots, i_P$ とおく．このサンプリングは 2．5 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 $j \in \{1, \cdots, D\}$ も以下の離散確率分布から P 回サンプリングする：

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

選ばれた添字を $j_1, j_2, \cdots, j_P$ 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：

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

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 $\mathbf{W} = \mathbf{U}_W\mathbf{\Sigma}_W\mathbf{V}_W{}^{\mathrm{T}}$ 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 $\mathbf{u}_1, \cdots, \mathbf{u}_K \in \mathbb{R}^P$ 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

$$\frac{\mathbf{U}_W(p, k)}{\mathbf{\Sigma}_W(k, k)\sqrt{P\mathcal{F}(i_p)}} \tag{5}$$

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，$\mathbf{S}^{\mathrm{T}}\mathbf{u}_k$ 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[6], [19]．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: $[\mathbf{W}, \{i_p\}_{p=1}^{P}] \leftarrow \mathrm{MatrixSampling}(\mathbf{X}, P)$
2: $[\mathbf{U}_W, \mathbf{\Sigma}_W, \mathbf{V}_W] \leftarrow \mathrm{SVD}(\mathbf{W})$
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 $\hat{\mathbf{u}}_k \in \mathbb{R}^P$ $(k = 1, \cdots, K)$ を定義
5: グラム・シュミットの正規直交化法を $\{\mathbf{S}^{\mathrm{T}}\hat{\mathbf{u}}_k\}_{k=1}^{K}$ に適用し，$\{\mathbf{S}^{\mathrm{T}}\mathbf{u}_k\}_{k=1}^{K}$ が正規直行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 [10] 参照）

2.3 正準相関分析

本報告で取り扱うもう一つの統計手法，正準相関分析のアルゴリズムを説明する．正準相関分析はサンプル数の等しい二つのデータ行列 $\mathbf{X} \in \mathbb{R}^{N \times D_1}, \mathbf{Y} \in \mathbb{R}^{N \times D_2}$ が与えられたとき，二つに共通する成分を線形変換で抽出する手法である．その共通する成分を $\mathbf{X}, \mathbf{Y}$ から抽出するための重みベクトルをそれぞれ $\hat{\mathbf{w}}_X \in \mathbb{R}^{D_1}, \hat{\mathbf{w}}_Y \in \mathbb{R}^{D_2}$ とおく．これらは以下の最適化問題を解くことで与えられる：

$$\hat{\mathbf{w}}_X, \hat{\mathbf{w}}_Y = \underset{\mathbf{w}_X, \mathbf{w}_Y}{\mathrm{argmax}}\ \mathrm{corr}\left[\mathbf{X}\mathbf{w}_X, \mathbf{Y}\mathbf{w}_Y\right]. \tag{6}$$

ここで，$\mathrm{corr}[\cdot, \cdot]$ は二つのベクトルを引数に，その相関係数を返す関数を表す．$\mathbf{X}\hat{\mathbf{w}}_X, \mathbf{Y}\hat{\mathbf{w}}_Y$ が抽出された一番目の共通成分となる．第二番目以降の共通成分も上記と類似の最適化問題を解くことで与えられるが，本報告では定義の詳細は割愛する（詳細については [10], [20] などを参照）．第 k 番目の共通成分を得るための重みベクトルを $\hat{\mathbf{w}}_X^{(k)}, \hat{\mathbf{w}}_Y^{(k)}$ とおく．通常のデータ解析では，上位の共通成分のみに興味があることが多い．ここでは上位 K 個の共通成分に興味があるとし，その重みベクトルを行列としてまとめ，$\mathbf{W}_X = \left[\hat{\mathbf{w}}_X^{(1)}, \cdots, \hat{\mathbf{w}}_X^{(K)}\right], \mathbf{W}_Y = \left[\hat{\mathbf{w}}_Y^{(1)}, \cdots, \hat{\mathbf{w}}_Y^{(K)}\right]$ とおく．この行列は特異値分解を用いることで計算できることが知られている [21]．ここでは詳細は割愛し，以下にそれを計算するアルゴリズムを擬似コードとしてまとめる．

アルゴリズム 3: 正準相関分析
Input: データ行列 $\mathbf{X} \in \mathbb{R}^{N \times D_1}$, $\mathbf{Y} \in \mathbb{R}^{N \times D_2}$, パラメータ K
Output: 重みベクトルの行列 $\mathbf{W}_X \in \mathbb{R}^{D_1 \times K}$, $\mathbf{W}_Y \in \mathbb{R}^{D_2 \times K}$
1: $[\mathbf{U}_1, \mathbf{\Sigma}_1, \mathbf{V}_1] \leftarrow \mathrm{SVD}(\mathbf{X})$
2: $[\mathbf{U}_2, \mathbf{\Sigma}_2, \mathbf{V}_2] \leftarrow \mathrm{SVD}(\mathbf{Y})$
3: $[\mathbf{U}_3, \mathbf{\Sigma}_3, \mathbf{V}_3] \leftarrow \mathrm{SVD}(\mathbf{U}_1{}^{\mathrm{T}}\mathbf{U}_2)$
4: $\mathbf{W}_X \leftarrow \mathbf{V}_1\mathbf{\Sigma}_1{}^{-1}\mathbf{U}_3(:, 1{:}K)$
5: $\mathbf{W}_Y \leftarrow \mathbf{V}_2\mathbf{\Sigma}_2{}^{-1}\mathbf{V}_3(:, 1{:}K)$

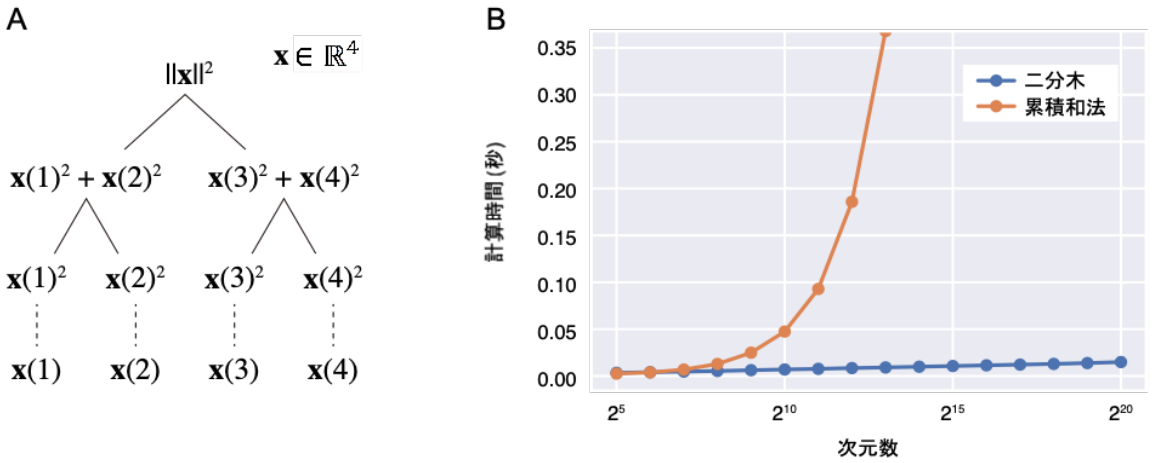

図 1. 二分木データ構造とそれを用いたサンプリング．**(A)** 二分木構造を用いたデータ構造．**(B)** サンプリングに必要な計算時間．二分木を用いた場合と累積和法を用いた方法をそれぞれ 10 回ずつ行い，その計算時間の平均を次元数の関数としてプロットした．

2.4 量子インスパイア正準相関分析

正準相関分析のアルゴリズムは特異値分解に基づくため，その特異値分解を量子インスパイア特異値分解に置き換えることで，高速化が可能である．本報告では Koide-Majima & Majima [10] で用いられた量子インスパイア正準相関分析のアルゴリズムと同一のものを用い実験を行った．アルゴリズム導出の詳細は本報告では割愛し，以下にその擬似コードをまとめる．

アルゴリズム 4: 量子インスパイア正準相関分析

Input: データ行列 $\mathbf{X} \in \mathbb{R}^{N \times D_1}$, $\mathbf{Y} \in \mathbb{R}^{N \times D_2}$,
パラメータ K, L, P

Output: Description $\{\mathbf{A}^{(1)}, \{\mathbf{w}_{l_1}^{(1)}\}_{l_1=1}^{L_1}\}$, $\{\mathbf{A}^{(2)}, \{\mathbf{w}_{l_2}^{(2)}\}_{l_2=1}^{L_2}\}$

1: $\left[\mathbf{S}^{(1)}, \{\mathbf{u}_l^{(1)}\}_{l=1}^{L}, \{i_p^{(1)}\}_{p=1}^{P}\right] \leftarrow \mathrm{qiSVD}(\mathbf{X}^{\mathrm{T}}, L, P)$

2: $\left[\mathbf{S}^{(2)}, \{\mathbf{u}_l^{(2)}\}_{l=1}^{L}, \{i_p^{(2)}\}_{p=1}^{P}\right] \leftarrow \mathrm{qiSVD}(\mathbf{Y}^{\mathrm{T}}, L, P)$

3: $[\mathbf{U}_3, \mathbf{\Sigma}_3, \mathbf{V}_3] \leftarrow$
$\mathrm{SVD}\left(\left[\mathbf{u}_1^{(1)}, \mathbf{u}_2^{(1)}, \cdots, \mathbf{u}_L^{(1)}\right]^{\mathrm{T}} \mathbf{S}^{(1)} \mathbf{S}^{(2)^{\mathrm{T}}} \left[\mathbf{u}_1^{(2)}, \mathbf{u}_2^{(2)}, \cdots, \mathbf{u}_L^{(2)}\right]\right)$

4: $\left[\mathbf{w}_1^{(1)}, \cdots, \mathbf{w}_K^{(1)}\right] \leftarrow \left[\mathbf{u}_1^{(1)}, \cdots, \mathbf{u}_L^{(1)}\right] \mathbf{U}_3(:, 1{:}K)$

5: $\left[\mathbf{w}_1^{(2)}, \cdots, \mathbf{w}_K^{(2)}\right] \leftarrow \left[\mathbf{u}_1^{(2)}, \cdots, \mathbf{u}_L^{(2)}\right] \mathbf{V}_3(:, 1{:}K)$

6: $\mathbf{A}^{(1)} \in \mathbb{R}^{P \times D_1}$ を (p, d) 番目の要素が以下となる行列として定義

$$\mathbf{A}^{(1)}(p,d) = \begin{cases} 1 & (i_p^{(1)} = d) \\ 0 & (\text{otherwise}) \end{cases}$$

7: $\mathbf{A}^{(2)} \in \mathbb{R}^{P \times D_2}$ を (p, d) 番目の要素が以下となる行列として定義

$$\mathbf{A}^{(2)}(p,d) = \begin{cases} 1 & (i_p^{(2)} = d) \\ 0 & (\text{otherwise}) \end{cases}$$

2.5 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・正準相関分析はいずれも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 （正準相関分析の場合は D_1 と D_2）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・正準相関分析の次元数に対する計算量はそれぞれ $O(\log(D))$, $O(\log(D_1 D_2))$ 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：

$$\frac{1}{\|\mathbf{x}\|^2} \sum_{n=1}^{n'-1} \mathbf{x}(n)^2 \leq u < \frac{1}{\|\mathbf{x}\|^2} \sum_{n=1}^{n'} \mathbf{x}(n)^2 . \qquad (7)$$

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は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 $\mathbf{X} = \mathbf{ZB}$ 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 $\{2^5, 2^6, \cdots 2^{15}\}$ で変化させた．

正準相関分析，量子インスパイア正準相関分析の評価には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 $\mathbf{Z} \in \mathbb{R}^{N\times 100}$, $\mathbf{B}_1 \in \mathbb{R}^{100\times D_1}$, $\mathbf{B}_2 \in \mathbb{R}^{100\times D_2}$, $\mathbf{E}_1 \in \mathbb{R}^{N\times D_1}$, $\mathbf{E}_2 \in \mathbb{R}^{N\times D_2}$ を作る．そして，データ行列 $\mathbf{X} \in \mathbb{R}^{N\times D_1}$, $\mathbf{Y} \in \mathbb{R}^{N\times D_2}$ を以下として作成した：

$$\mathbf{X} = \mathbf{ZB}_1 + 0.5\mathbf{E}_1, \qquad \mathbf{Y} = \mathbf{ZB}_2 + 0.5\mathbf{E}_2 .$$

上記の生成手続きは確率正準相関分析の生成モデルの仮定と一致し，正準相関分析の評価としては標準的に用いられる形のものである．主成分分析の場合と同様，$N = 10000$ とし，データの次元数 D は $\{2^5, 2^6, \cdots 2^{15}\}$ で変化させた．

3.2 実ベンチマークデータ

主成分分析，量子イン本報告では実データとして主成分分析，正準相関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 [10] と同様の手続きで行なった．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 [22]，データセット 2 は CIFAR10 [23]である．主成分分析，量子インスパイア主成分分析の評価ではピクセル値を特徴量とみなし，両アルゴリズムを適用した．正準相関分析，量子インスパイア正準相関分析の評価では，各画像を左半分と右半分に分割し，それをデータのペアとみなし，両アルゴリズムを適用した．

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 [24]，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 [25]である．いずれも文を wikipedia2vec [26] という手法により 300 次元のベクトルに変換してから用いた．主成分分析，量子インスパイア主成分分析の評価では，英語側 300 次元のみを用い，両アルゴリズムを適用した．正準相関分析，量子インスパイア正準相関分析の評価では，対訳をデータのペアとみなし，両アルゴリズムを適用した．

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 [27] 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．主成分分析，量子インスパイア主成分分析の評価では音声側のデータのみを用い，正準相関分析，量子インスパイア正準相関分析の評価では，音声-筋電の対応を対とするデータとみなし，アルゴリズムを適当した．

3.3 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：

$$1 - \frac{\left\|\mathbf{X} - \mathbf{XWW}^{\mathrm{T}}\right\|_F^2}{\|\mathbf{X}\|_F^2} . \tag{8}$$

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 $[\widehat{\mathbf{w}}^{(1)}, \widehat{\mathbf{w}}^{(2)}, \cdots, \widehat{\mathbf{w}}^{(K)}]$ である．上記の評価には訓練データとテストデータは分離せず，同一のデータ行列を用いて重みベクトルの計算と評価を行った．

正準相関分析，量子インスパイア正準相関分析の評価では，上位 100 個の共通成分を抽出した．抽出した 100 個の各成分に関して $\mathbf{X}$ 側から抽出した値と $\mathbf{Y}$ 側から抽出した値の相関係数を求め，上位 K 成分の相関係数の和を評価基準とした．これは先行研究 Andrew et al. [17]で用いられている評価方法と同じものである．

4. 実験結果

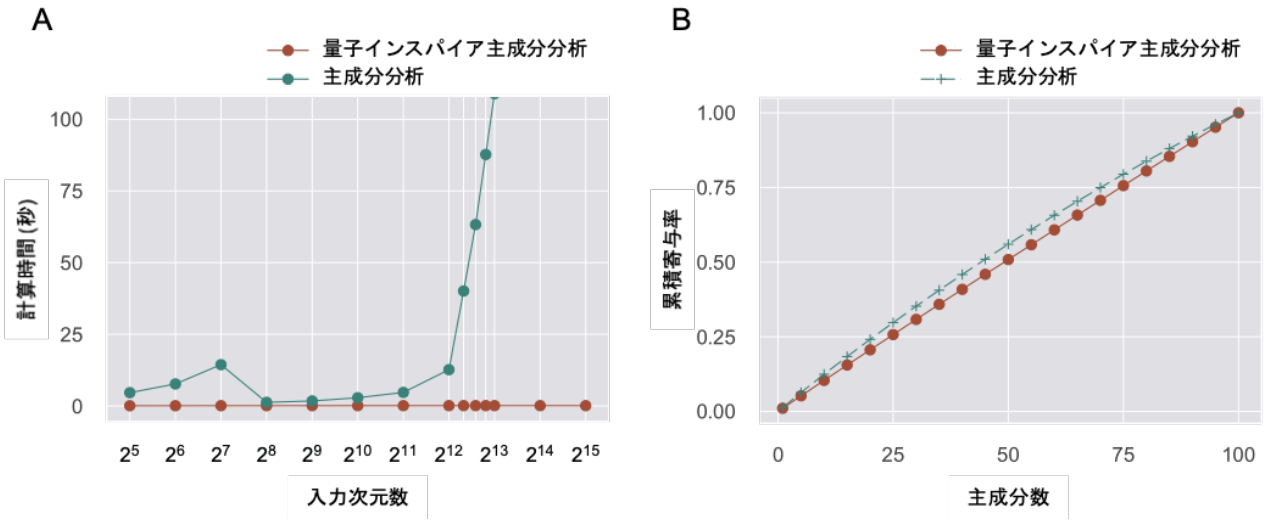

図 2. 主成分分析と量子インスパイア主成分分析のシミュレーションデータを用いた比較. **(A)** 計算時間の比較. 入力次元数を変えつつ，10 回の平均計算時間をプロットした. **(B)** 累積寄与率による性能比較. 入力次元数を 10000 に固定し，主成分数の関数として累積寄与率をプロットした.

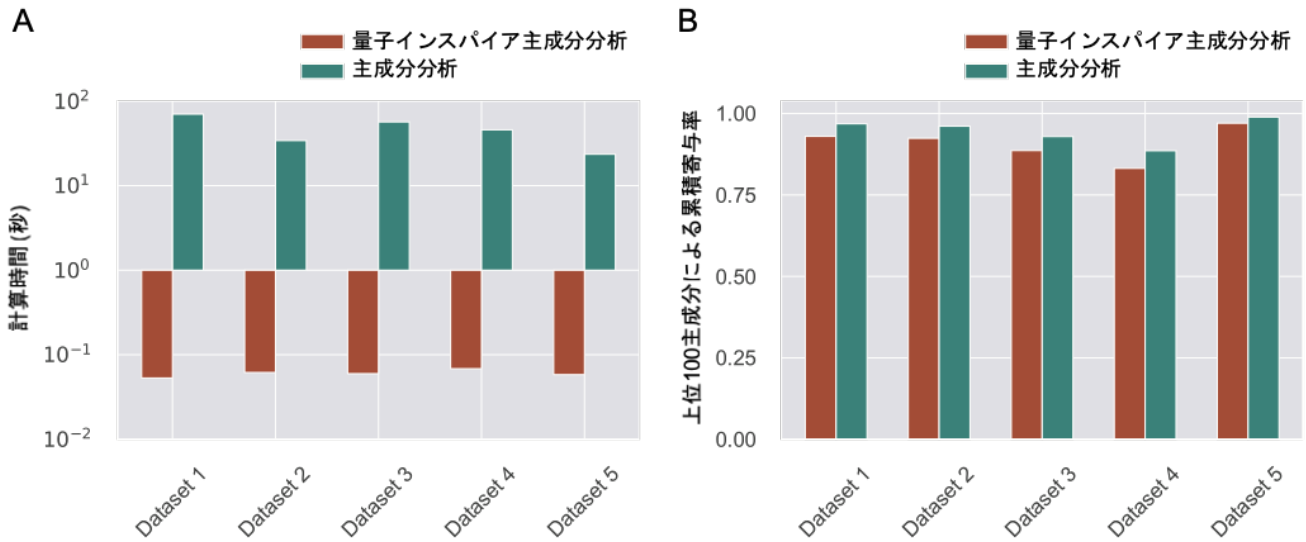

図 3. 主成分分析と量子インスパイア主成分分析の実データを用いた比較. **(A)** 計算時間の比較. 10 回の平均計算時間をプロットした. **(B)** 累積寄与率による性能比較. 上位 100 主成分を抽出し，その累積寄与率を評価した.

4.1 量子インスパイア主成分分析の評価

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そ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.

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.

4.2 量子インスパイア正準相関分析の評価

次にシミュレーションデータを用い，正準相関分析と量子インスパイア正準相関分析の計算時間を比較した. 主成分分析での実験と同様，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 4A）. 主成分分析での実験結果と同様，量子インスパイア正準相関分析による計算時間の改善が確認された. また，性能に関しての評

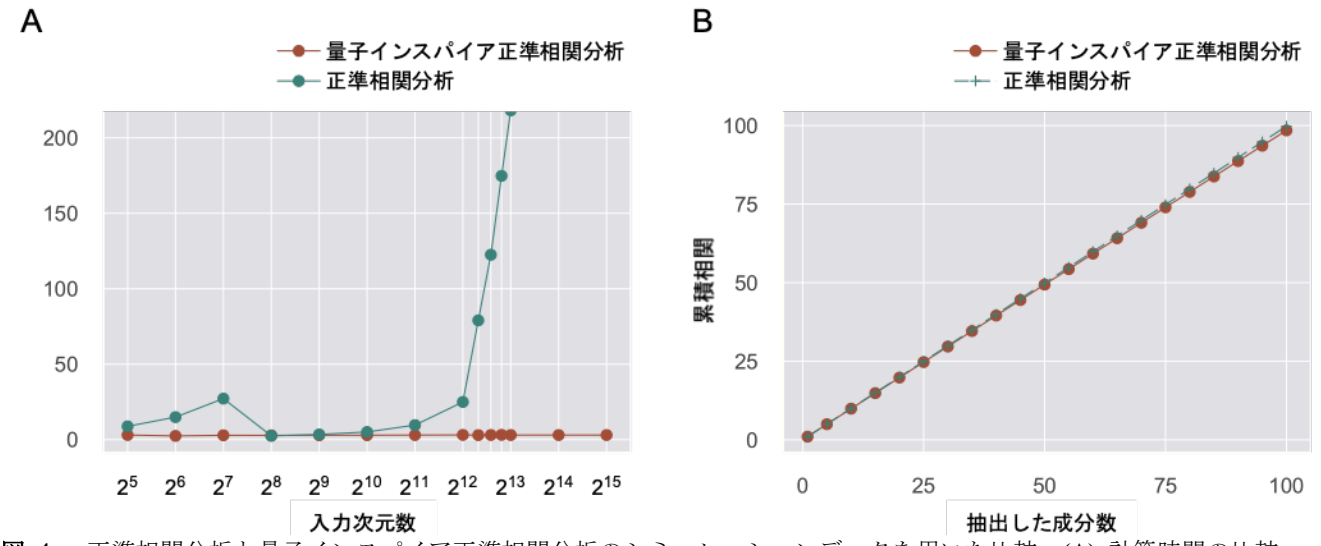

図 4. 正準相関分析と量子インスパイア正準相関分析のシミュレーションデータを用いた比較．**(A)** 計算時間の比較．入力次元数を変えつつ，10 回の平均計算時間をプロットした．**(B)** 累積相関による性能比較．入力次元数を 10000 に固定し，抽出した共通成分の数の関数として累積相関をプロットした．

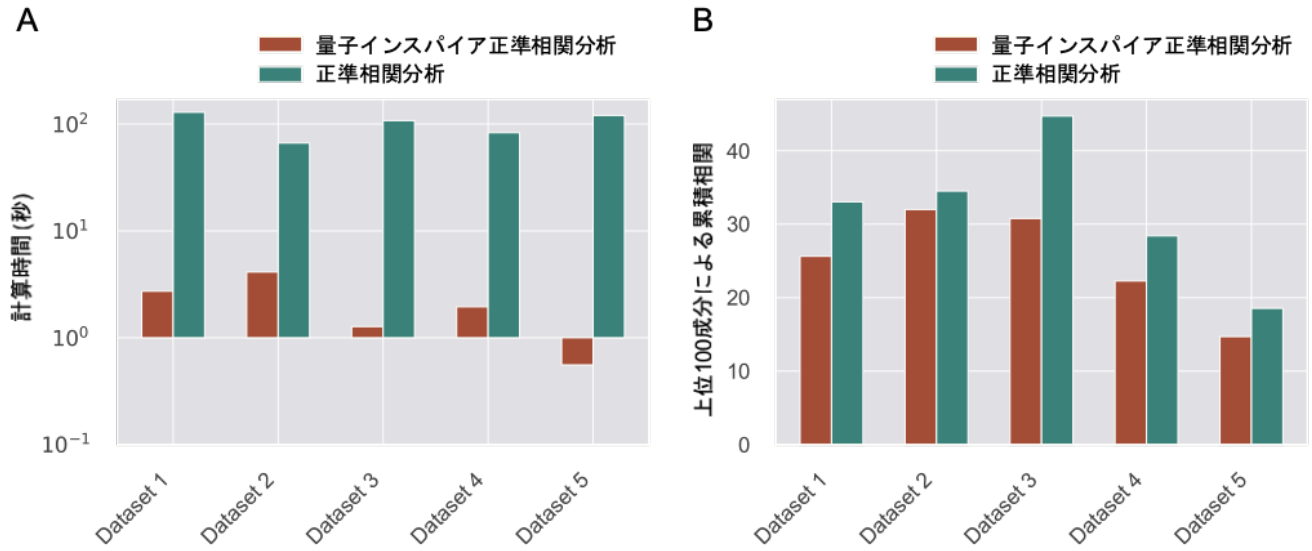

図 5. 正準相関分析と量子インスパイア正準相関分析の実データを用いた比較． **(A)** 計算時間の比較．10 回の平均計算時間をプロットした．**(B)** 累積相関による性能比較．上位 100 の共通成分を抽出し，その累積相関を評価した．

価を上位 k 個の共通成分によって抽出された相関係数の和（3．3 節参照）で評価した（図 4B）．結果，性能の低下は 1%以下であった．

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 5A,B）．正準相関分析と比較し，量子インスパイア正準相関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 7–13%であった．

4.3 量子インスパイア正準相関分析による非線形相関抽出

量子インスパイア正準相関分析による高速計算が，単なる計算時間の短縮だけでなく，よりよい共通成分の抽出に役立つ一例を示す．ここまでの実験では共通成分の抽出を線形変換に限定していた．しかし，カーネル正準相関分析 [13]–[16] や深層正準相関分析 [17] では，カーネル，深層ニューラルネットワークによる非線形変換を用いてより高く相関する共通成分を得られることが知られている．非線形変換が有用であることから，ここでは 2 次の多項式による非線形変換と量子インスパイア正準相関分析を組み合わせた方法を提案する．

具体的には，データ行列内の D 個の変量に対し，変量の全ペア間で積をとり，得られた ${}_DC_2$ 個の積の値を新たな変量として横に並べたデータ行列を作成する．この操作を対として与えられる 2 つのデータ行列 $\mathbf{X}, \mathbf{Y}$ のそれぞれに対

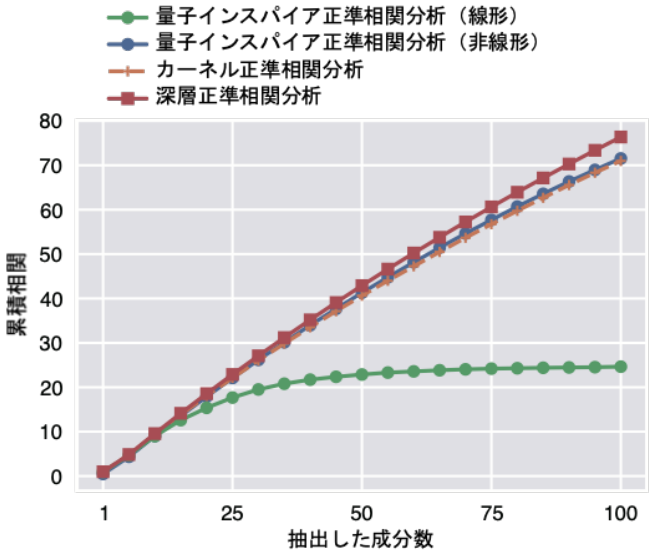

図 6. 正準相関分析の派生手法の性能比較．データセット 1 を用い，抽出した共通成分の数の関数として累積相関をプロットした．

して行う．この非線形変換によって得られたデータ行列に対して，量子インスパイア正準相関分析を行った．この操作はデータの次元数（データ行列の列数）を D 個から $D + 0.5D(D+1)$ に増加させる．次元数が大きく増加するため，通常の正準相関分析を用いた場合は計算時間の問題から処理することができない．

上記の提案法を実データセット 1 に適用し，抽出された相関を評価した（図 6）．比較対象として，線形変換のみに基づく量子インスパイア正準相関分析，カーネル正準相関分析，深層正準相関分析を用いた．なお，本実験では，（正準相関分析と量子インスパイア正準相関分析との間の比較ではなく）他の非線形機械学習手法との性能比較に興味があるため，訓練データとテストデータを分け，汎化性能を評価した．結果，提案した方法は線形変換のみに基づく正準相関分析に比べ，多くの相関を抽出し，その性能はーネル正準相関分析，深層正準相関分析に匹敵するものであった．

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・量子インスパイア正準相関分析の計算時間，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において評価した．結果，性能低下を伴うものの，数千万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%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高次元データの解析を可能にする

また，非線形変換と組み合わせることで，量子インスパイア正準相関分析がカーネル相関分析や深層相関分析に匹敵する性能（相関抽出能力）を発揮できる例が示された．この結果は，量子インスパイアアルゴリズムが単なる計算時間短縮だけでなく，新しい非線形統計手法の開発につながる可能性を示している．

謝辞

本研究は JSPS 科研費 JP20K16465 の助成を受けたものです。

参考文献

[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, doi: 10.1038/nn.3776.

[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, doi: 10.1371/journal.pcbi.1006907.

[3] N. Koide-Majima, T. Nakai, and S. Nishimoto, “Distinct dimensions of emotion in the human brain and their representation on the cortical surface,” *Neuroimage*, vol. 222, p. 117258, Aug. 2020, doi: 10.1016/j.neuroimage.2020.117258.

[4] N. Yahata *et al.*, “A small number of abnormal brain connections predicts adult autism spectrum disorder,” *Nat Commun*, vol. 7, p. 11254, Apr. 2016, doi: 10.1038/ncomms11254.

[5] H.-T. Wang *et al.*, “Finding the needle in a high-dimensional haystack: Canonical correlation analysis for neuroscientists,” *Neuroimage*, p. 116745, Apr. 2020, doi: 10.1016/j.neuroimage.2020.116745.

[6] 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, doi: 10.1145/3313276.3316310.

[7] 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, Accessed: Jun. 23, 2019. [Online]. Available: http://arxiv.org/abs/1811.04852.

[8] 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, Accessed: Jun. 23, 2019. [Online]. Available: http://arxiv.org/abs/1811.04909.

[9] E. Tang, “Quantum-inspired classical algorithms for principal component analysis and supervised clustering,” *arXiv:1811.00414 [quant-ph]*, Oct. 2018, Accessed: Jun. 23, 2019. [Online]. Available: http://arxiv.org/abs/1811.00414.

[10] N. Koide-Majima and K. Majima, “Quantum-inspired canonical correlation analysis for exponentially large dimensional data,” *Neural Netw*, vol. 135, pp. 55–67, Mar. 2021, doi: 10.1016/j.neunet.2020.11.019.

[11] 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, Accessed: May 03, 2020. [Online]. Available: http://arxiv.org/abs/1907.05568.

[12] C. Ding, T.-Y. Bao, and H.-L. Huang, “Quantum-Inspired Support Vector Machine,” *arXiv:1906.08902 [quant-ph, stat]*, Jul. 2019, Accessed: May 03, 2020. [Online]. Available: http://arxiv.org/abs/1906.08902.

[13] S. Akaho, “A kernel method for canonical correlation analysis,” 2001.

[14] F. R. Bach and M. I. Jordan, “Kernel Independent Component Analysis,” *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[15] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical Correlation Analysis: An Overview with Application to Learning Methods,” *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004, doi: 10.1162/0899766042321814.

[16] T. Melzer, M. Reiter, and H. Bischof, “Nonlinear Feature Extraction Using Generalized Canonical Correlation Analysis,” in *Artificial Neural Networks — ICANN 2001*, 2001, pp. 353–360.

[17] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, “Deep Canonical Correlation Analysis,” in *Proceedings of the 30th International Conference on Machine Learning*, Feb. 2013, pp. 1247–1255, [Online]. Available: http://proceedings.mlr.press/v28/andrew13.html.

[18] K. Majima *et al.*, “Decoding visual object categories from temporal correlations of ECoG signals,” *Neuroimage*, vol. 90, pp. 74–83, Apr. 2014, doi: 10.1016/j.neuroimage.2013.12.020.

[19] 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, doi: 10.1109/SFCS.1998.743487.

[20] D. Chu, L.-Z. Liao, M. K. Ng, and X. Zhang, “Sparse Canonical Correlation Analysis: New Formulation and Algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 3050–3065, Dec. 2013, doi: 10.1109/TPAMI.2013.104.

[21] W. H. Press, “Canonical Correlation Clarified by Singular Value Decomposition.” 2011, [Online]. Available: http://numerical.recipes/whp/notes/CanonCorrBySVD.pdf.

[22] 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, doi: 10.1109/5.726791.

[23] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” p. 60, 2009.

[24] 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, doi: 10.3115/v1/P14-2080.

[25] R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz, “JESC: Japanese-English Subtitle Corpus,” *arXiv:1710.10639 [cs]*, 2017, Accessed: Jun. 07, 2020. [Online]. Available: http://arxiv.org/abs/1710.10639.

[26] 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, Accessed: Jun. 07, 2020. [Online]. Available: http://arxiv.org/abs/1812.06280.

[27] 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.