

一般発表 | Track 4: メディア処理・HCI・人間中心情報マネジメント

2026年2月28日(土) 9:30 ~ 11:40 | 会場

[1H] 可視化

座長:梅本 和俊(SB Intuitions株式会社) コメントータ:北山 大輔(工学院大学) ジュニアコメントータ:周 文
トウ(九州大学)

9:30 ~ 9:55

[1H-01] LiDARと骨格推定AIを用いた投球動作の上達補助フィードバックシステム

*石倉 朝陽¹、秋岡 明香¹ (1. 明治大学)

9:55 ~ 10:20

[1H-02] SQLを拡張した情報可視化言語に対するGUI上での可換性の設計およびライフログ分析
支援の有効性評価

*宇津木 萌¹、大久保 勇輝¹、川原 一輝¹、富井 尚志¹ (1. 横浜国立大学)

10:20 ~ 10:45

[1H-03] ファイチュeringの最適化支援のための埋め込み移動軌跡の可視化分析

*石川 智也¹、藤田 秀之¹、大森 匡¹、新谷 隆彦¹ (1. 電気通信大学)

10:45 ~ 11:10

[1H-04] 不動産情報探索VRインタフェースにおける属性分布可視化

*中山 裕紀¹、大島 裕明¹ (1. 兵庫県立大学)

LiDAR と骨格推定 AI を用いた投球動作の上達補助 フィードバックシステム

石倉 朝陽[†] 秋岡 明香[†]

[†] 明治大学総合数理学部 〒168-8525 東京都中野区中野 4-21-1

E-mail: {ev211045. akioka}@meiji.ac.jp

あらまし 野球の投球技術向上において、データに基づく客観的なフォーム解析への需要が高まっている。しかし、高精度な3次元動作計測は高価な専用機材を必要とし、一方で安価な単眼カメラ解析では、算出された数値データ（肘の角度やストライド長など）を具体的な身体操作へ変換することが困難であるという「解釈の壁」が課題となっている。そこで本研究では、モバイル端末と骨格推定 AI を統合し、屋外環境下で手軽に3次元動作解析を行うシステムを提案する。本手法は、投手の過去の好調時データと現在の投球を比較（個人内比較）し、球速に最も寄与する動作因子を特定する。さらに、算出された数値差分を「靴半足分前へ」や「地面に突き刺す」といった、選手が直感的に理解可能な具体的なアドバイスへと自動変換して提示する点に独自性がある。実証実験の結果、本システムは投球指導において実用上妥当な計測精度を有することを確認した。また、提示されたフィードバックが未経験者の球速向上や、経験者のフォーム矯正に有効に機能することが示された。

キーワード 画像認識, フィードバック, スポーツ, 骨格推定, LiDAR

1. はじめに

近年、スポーツ分野におけるセンシング技術や機械学習の進展により、野球の投球動作においても客観的データに基づくフォーム解析への期待が高まっているが、既存の高精度な3次元解析は高価な設備を要し、一方で安価な単眼カメラによる2次元解析では奥行き情報の欠如や、算出された数値データ（肘の角度やストライド長など）を選手が具体的な身体操作へ直感的に変換できないという「解釈の壁」が課題となっている。そこで本研究では、LiDAR スキャナ搭載のモバイル端末を用いることで簡易かつ高精度な3次元動作計測を実現するとともに、個人の過去の好投時データとの比較（個人内比較）に基づき、差分数値を「靴半足分前へ」といった具体的かつ行動可能なアドバイスへと自動変換して提示するシステムを提案する。これにより、専門的なコーチが不在の環境下でも、選手が客観的データと身体感覚のギャップを埋め、自律的にフォーム改善に取り組める環境の構築を目指す。

2. 関連研究

本研究の基盤となる「骨格推定技術による動作解析」と、スポーツ指導における「フィードバック手法」の観点からこれらに関連する先行研究として以下の2つを挙げる。

2.1 映像に基づく骨格推定とスポーツ動作解析

近年、ディープラーニングの発展により、特殊なマ

ーカーを装着せずに映像のみから人物の関節位置を特定するマーカーレス骨格推定技術が急速に進展している。代表的な手法として、Cao らが提案した OpenPose[1]が挙げられる。OpenPose は、Part Affinity Fields (PAFs) と呼ばれる手法を用いることで、複数人の関節位置とそれらの繋がりをボトムアップに推定するアルゴリズムを確立し、高い精度での2次元骨格推定を実現した。この技術はスポーツ分野にも広く応用されており、投球フォームや走動作の解析において数多くの事例が報告されている。しかし、OpenPose の手法を含む多くの既存研究は、単眼 RGB カメラを用いた2次元平面上の解析に留まるか、あるいは高価な複数台カメラシステムや高性能 GPU を必要とするケースが大半である。投球動作のような奥行きを伴う複雑な3次元動作を、現場レベル（グラウンドやブルペン）で手軽かつ高精度に解析するには、単眼 RGB カメラの深度情報の欠如や、機材セットアップの煩雑さが障壁となっていた。本研究では、近年普及が進む LiDAR スキャナ搭載のモバイル端末を用いることで、単眼映像でありながら深度情報を取得し、簡便に3次元的な動作特徴（ストライドの物理的距離など）を計測する点に独自性がある。

2.2 運動学習におけるフィードバックの質

スポーツ技能の習得支援において、計測されたデータをどのように学習者に提示するかは、センシング技術と同様に重要な課題である。Sharma らによると運動学習の分野では、フィードバックは主に「結果の知識 (Knowledge of Results: KR)」と「パフォーマンスの知識 (Knowledge of Performance: KP)」の2つに大別され

る[2]. KRは「ボールがストライクゾーンに入ったか」「球速は何 km/h か」といった結果に関する情報であり, KPは「肘が下がっていた」「ストライドが狭かった」といった動作のプロセスや質に関する情報である. Sharma らの研究をはじめとする多くの先行研究において, 投球のような複雑なスキル習得には, 単なる結果 (KR) の提示だけでなく, 具体的な動作修正を促す KP の提示が不可欠であることが示唆されている. しかし, 従来の動作解析システムの多くは, 計測された KP (例: 肘の角度が 90 度, ストライドが 1.4m) を定量的な数値データとしてそのまま提示するに留まっている. 専門的な知識を持たない選手にとって, 数値化された KP を「具体的に体のどの部位をどう動かせばよいか」という身体感覚に翻訳することは容易ではない. 数値の羅列は, 学習者にとって認知的な負荷となり, 直感的な動作修正に結びつかない場合がある.

2.3 本研究のアプローチ

上述の課題に対し, 本研究は「計測の簡易化」と「フィードバックの具体化」の両立を目指すものである. 技術面では LiDAR 搭載端末を活用することで大規模機材を必要としない 3 次元計測環境を構築する. さらに指導面では, すでに重要性が示されている KP (動作情報) を, 単なる数値としてではなく, 「靴半足分前へ」「ボール 1 個分高く」といった, 選手が直感的に理解し行動に移せる具体的な指示へと自動変換してフィードバックする. これにより, データ分析の専門知識を持たない選手であっても, 客観データに基づいた自律的なフォーム改善が可能となるシステムを提案する.

3. 提案手法

3.1 提案手法の概要

本研究の全体処理フローを図 1 に示す. 本システムは, LiDAR 搭載端末と球速計を用いて取得した投球データに対し, 一連の解析処理を行うことで選手へのフィードバックを実現するものである. まず, 処理の初段では, RGB-D 映像から特定の開始・終了姿勢 (T ポーズ・Y ポーズ) を検出して投球区間を自動抽出し, 抽出された動作データから骨格と深度情報を統合して 3 次元的な運動特徴量を算出する (各ポイントの解析). 続いて, 得られたデータを球速に基づいて上位群と下位群に分類し, 両群間の特徴量の差異を個人内比較することで, パフォーマンスの変動に最も寄与する動作因子を特定する (影響因子の特定). 最終的に, 特定された因子の数値差分を「靴半足分」等の物理的目安や具体的な動作指示へと変換し, 選手が直感的に理解・実践可能な改善案として提示する (フィードバック).

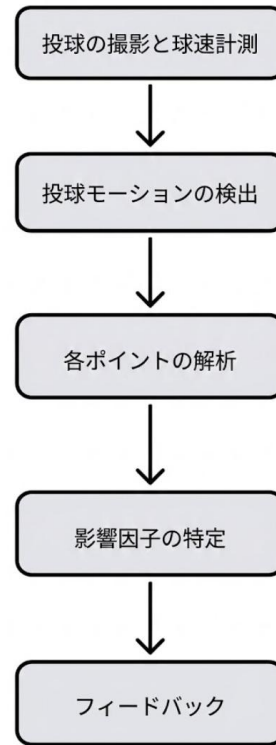


図 1 本研究のフローチャート

3.2 投球の撮影と球速計測

本システムにおける解析用データの取得には, LiDAR スキャナを搭載したモバイル端末と, IoT センサー内蔵ボールを用いる.

投球動作の映像および 3 次元形状データの取得には, iPhone17 Pro を使用する. 撮影用アプリケーションには Record3D[3]を用いる. Record3D は, iOS 端末の LiDAR センサーを用いて, 深度マップ (Depth Map) と RGB 映像をハードウェアレベルで同期して記録することが可能である. 本研究ではフレームレート 30fps の設定で撮影を行う.

投球パフォーマンスの客観的な指標として, 球速を用いる. 球速の測定には, 9 軸センサー (3 軸加速度, 3 軸地磁気, 3 軸角速度) を内蔵した IoT ボール「テクニカルピッチ」[4]を使用する. 本デバイスは, Bluetooth Low Energy (BLE) を介して計測データをモバイル端末へ送信する. 本研究では, このデバイスにより計測された球速値を, 投球の質を評価する基準として採用する.

投球映像と球速データは, 身体操作の習熟度が異なる 4 名の投手を対象に収集した. 被験者 A,C は野球経験者であり, 被験者 B,D は野球未経験者である. 日を変えて複数回撮影を行うことで収集した. 具体的には,

1 回あたり 10 球程度の投球を計測し、合計で約 50 球分のデータを蓄積している。このように複数日にわたってデータを集める目的は、一度きりの計測では捉えきれない「日ごとの調子の波」や、身体の使い方の微細な変化を分析に反映させ、結果の信頼性を高めるためである。

3.3 投球モーションの検出

3.3.1 YOLOv8m による骨格推定とキーポイント定義

投球動作の定量化には、映像内の人物の姿勢情報を高精度かつ高速に取得する必要がある。本研究では、骨格推定モデルとして Ultralytics 社が開発した YOLOv8-Pose[5]を採用した。YOLOv8 には計算コストと精度の異なる複数のモデル (n, s, m, l, x) が存在するが、本システムではモバイル端末等での処理速度と、複雑な投球動作を正確に捉える精度のバランスを考慮し、ミディアムモデルである「yolov8m-pose.pt」を使用した。本モデルによって検出される身体のキーポイント構成を図 2 に示す。鼻 (ID:0) から足首 (ID:16) までの計 17 点が検出され、各点は(x, y)座標と信頼度スコアを持つ。本研究では、特に投球動作に関与する肩、肘、手首などの上半身の点と、下半身の踏み込みや重心移動に関わる腰、膝、足首の座標を解析に利用する。表 2 には各身体のキーポイントの ID、部位そして、本研究における主な解析用途を示す。また、これらの座標計算や画像処理全般には、オープンソースライブラリである OpenCV[6]を用いた。

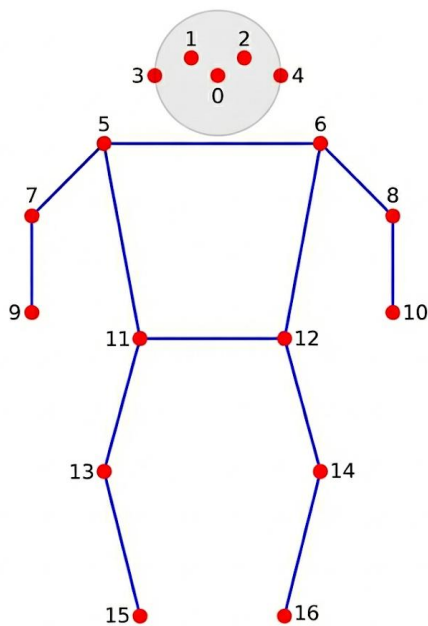


図 2 ランドマーク位置

表 1 ランドマーク対応表

ID	部位	本研究における主な解析用途
0	鼻	終了トリガー (Y ポーズ) の判定基準
1	左目	本解析では不使用
2	右目	本解析では不使用
3	左耳	本解析では不使用
4	右耳	本解析では不使用
5	左肩	開始トリガー (T ポーズ) 判定, 体幹の前傾, 肘伸展の算出
6	右肩	開始トリガー (T ポーズ) 判定, 体幹の前傾, 肘伸展の算出
7	左肘	肘の伸展度合い (リリース時) の算出 (左投げの場合)
8	右肘	肘の伸展度合い (リリース時) の算出 (右投げの場合)
9	左手首	開始・終了トリガー, リリース特定, 高さ・伸展算出
10	右手首	開始・終了トリガー, リリース特定, 高さ・伸展算出
11	左腰	前足のブロックング, 体幹の前傾, 深度補正の参照
12	右腰	前足のブロックング, 体幹の前傾, 深度補正の参照
13	左膝	前足のブロックング算出, 深度情報の補正参照 (前足時)
14	右膝	前足のブロックング算出, 深度情報の補正参照 (前足時)
15	左足首	ストライド幅, 前足のブロックング, 地面基準点の特定
16	右足首	ストライド幅, 前足のブロックング, 地面基準点の特定

3.3.2 ポーズ定義による動作区間の抽出

実際の投球映像から、セットポジションへの移動や待機時間を排除し、投球動作のみを自動で切り出すために、本システムでは以下の 2 つのポーズを動作のトリガーとして定義している。

開始トリガー (T ポーズ): 図 3 (左) に示すように、投手が直立し、両手を水平に広げた姿勢である。YOLO で検出された両肩 (ID:5, 6) と両手首 (ID:9, 10) の Y 座標がほぼ一致した状態が一定フレーム続いた時点を、投球モーションの開始とする。

終了トリガー (Y ポーズ): 図 3 (右) に示すように、投球終了後に両手を頭より高く挙げた姿勢である。両手首 (ID:9, 10) の Y 座標が鼻 (ID:0) の Y 座標より

も上にある状態を検出し、これをモーションの終了とする。システムは映像を監視し、Tポーズ検知からYポーズ検知までの区間を1試行分のデータとして自動保存する。



図 3 Tポーズ（左），Yポーズ（右）姿勢

3.3.3 リリースフレームの特定

投球パフォーマンスにおいて最も重要な瞬間である「ボールリリース」の特定には、投球腕の手首キーポイント（右投手であれば ID:10）の座標推移を利用する。具体的には、切り出されたモーション区間内において手首の座標を追跡し、投球方向への移動速度が最大となる瞬間をリリースフレームとして定義する。特定されたリリース時のフレーム例を図4に示す。次節で述べるストライドや肘角度などの特徴量は、この瞬間の各キーポイント座標に基づいて算出される。



図 4 リリースフレームの RGB 画像（左）と深度マップ（右）

3.4 各ポイントの解析

本節では、各投球試行における5つの動作特徴量（ストライド、リリースの高さ、前足のブロッキング、肘

の伸展、体幹の前傾）を算出する。特定されたリリースフレームにおける投球動作の定量的評価指標の算出手法について述べる。

3.4.1 座標系の定義と算出ロジック

YOLOv8-Pose によって取得されるキーポイント座標は、画像平面上の2次元座標 $p(u, v)$ である。本システムでは、評価項目の特性に応じて、深度情報を用いた3次元座標解析と、画像平面上の2次元座標解析を使い分けるハイブリッドな手法を採用する。

3.4.2 評価指標の定義と算出ロジック

投球パフォーマンスに寄与する以下の5つの指標について、それぞれの算出ロジックを定義する。

3次元座標への変換

ストライドやリリースの高さなど、物理的な距離計測が必要な項目については、Record3Dより取得した深度マップ（Depth Map）を利用して3次元空間座標 $P(X, Y, Z)$ へ変換する。画像上のキーポイント座標 (u, v) に対応する深度値 d 、カメラの焦点距離 (f_x, f_y) 、および光学中心 (c_x, c_y) を含む内部パラメータは、Record3Dが出力するメタデータより取得した値を用いる。カメラ座標系における3次元位置は以下のピンホールカメラモデル式により算出される。

$$\begin{cases} Z = d \\ X = (u - c_x) \times Z / f_x \\ Y = (v - c_y) \times Z / f_y \end{cases}$$

2次元座標の利用

一方、身体の角度評価において、図4（右）のようにLiDARの性質上、素早く動くものの深度を捉えられず、右肩から投げ手にかけての奥行き情報が欠如してしまい深度値の信頼性が低い部位については2次元座標 (u, v) をそのまま用いる。

評価指標の定義

投球パフォーマンスに寄与する以下の5つの指標について、それぞれの算出ロジックを定義する。

I. ストライド [3次元算出]

踏み出し足の接地位置と軸足（プレート板）との物理的な距離である。踏み出し足の足首 P_{lead} と軸足の足首 P_{trail} の3次元座標間のユークリッド距離を以下の式をもとに算出する。

$$D_{stride} = \|P_{lead} - P_{trail}\|_2$$

II. 前足のブロッキング [3次元算出]

着地した前足の膝の伸展度合いである。踏み出し足の「腰」、「膝」、「足首」の3点座標からなる2つのベクトルを定義し、3次元空間上で内角を算出する。

III. リリースの高さ [3次元算出]

ボールを離す瞬間の打点の高さである。踏み出し足の足首座標 P_{lead} を地面の基準点とし、投球腕手首座標 P_{wrist} との鉛直方向（Y軸成分）の差分 $Y_{wrist} - Y_{lead}$ とし

て算出する。また図 4 右のように投球時は投げ手の肩から手にかけて深度情報が欠落しているため、反対の肩の深度を用いて差分を算出する。

IV. 肘の伸展 [2次元算出]

リリース時における肘の伸び具合を示す指標である。投球腕の「肩」、「肘」、「手首」の2次元座標(u, v)を用いる。肘関節を始点とし、肩関節へ向かう上腕ベクトル v_{upper} と手首関節へ向かう前腕ベクトル v_{fore} のなす角 θ を、内積公式を用いて算出する。

$$\cos \theta = \frac{v_{upper} \times v_{fore}}{|v_{upper}| |v_{fore}|}$$

V. 体幹の前傾[2次元算出]

リリース時に上体が打者方向へどれだけ倒れ込んでいるかを示す指標である。2次元画像上において、腰の中心点と首を結ぶ体幹ベクトルと、画像の垂直軸(Y軸)とのなす角度を算出する。

3.4.3 データの統合とデータセットの構築

最終的な分析を行うために、これらの動作データと、3.2節で述べたテクニカルピッチによって計測された球速データを統合する。具体的には、同一試行(投球)における映像解析結果と球速値を、投球順序およびタイムスタンプに基づいて紐付けを行う。この統合処理により、1行が1つの投球試行に対応し、列に「正解ラベル(球速)」および「説明変数(5つの動作特徴量)」を持つ構造化データを作成する。作成されたデータセットはCSV(Comma-Separated Values)形式で出力され、次章における影響因子の特定および統計的比較のための入力データとして用いられる。

3.5 影響因子の特定

本節では、構築されたCSVデータセットを用いて、その投手の球速に最も寄与している動作因子、あるいは球速低下の原因となっている動作の差異を特定するアルゴリズムについて述べる。

3.5.1 パフォーマンスの分類基準

一般的な動作解析では、プロ選手の平均値などを理想基準とすることが多いが、体格や筋力の異なる他者との比較は必ずしも有効ではない。そこで本研究では、選手自身の過去データを用いた個人内比較を行う。具体的には、各撮影日(セッション)の投球データを球速の降順にソートし、上位10%の試行群を「好調時(Good)」、残りの90%を下位群「通常時(Standard)」として定義する。上位10%をその選手における当該セッションの「正解」と仮定し、両群の動作特徴量の差異を分析することで、好調時の身体操作を探索する。

3.5.2 影響度の算出と特定

好調時(Good)と通常時(Standard)の差異を定量化するために、各特徴量 i (ストライド、リリース高さ等)について、両群の平均値 μ_{good}^i および μ_{std}^i を算出す

る。続いて、以下の式を用いて通常時に対する好調時の変化率(影響度スコア)を特徴量ごとに計算する。

$$Score_i = \left| \frac{\mu_{good}^i - \mu_{std}^i}{\mu_{std}^i} \right| \times 100$$

システムは、3章で定義した5つの特徴量の中でこの $Score_i$ が最大となる項目を、その日の投球パフォーマンスを左右する「最重要因子」として特定する。

3.5.3 日ごとの傾向分析と統合診断

本システムでは、全日程のデータを単純に合算して解析するのではなく、まず撮影日ごとに独立して影響因子の特定を行う。これは、日によって身体のコンディションや球速のベースラインが異なるため、単純なデータの合算では解析精度が低下する恐れがあるためである。その上で、最終的な課題特定においては、各撮影日で選出された重要因子の出現頻度に基づく、多数決的な統合診断を行う。例えば、3日間の計測データに対し、A日の分析では「ストライド」、B日では「肘の伸展」、C日では「ストライド」がそれぞれ重要因子として特定されたとする。この場合、日ごとの変動はあるものの、全体傾向としては「ストライド」が最も頻繁にパフォーマンスに影響を与えていると判断し、これをその投手の根本的な改善課題として特定する。この二段階のプロセスを経ることで、その日限りの偶発的な調子の変動を除外し、投手が恒常的に抱えている技術的ボトルネックを抽出することが可能となる。

3.6 フィードバック手法

本システムの核心は、算出された数値データをそのまま提示するのではなく、選手が直感的に修正動作を想起できる「具体的アドバイス」へ変換する点にある。特定された重要因子について、球速上位群(Good群)に属する全試行における当該動作特徴量(ストライド長や膝角度等)の平均値を μ_{fast} と、下位群(Standard群)における同特徴量の平均値を μ_{slow} の差分 $\Delta = \mu_{fast} - \mu_{slow}$ を算出する。この Δ の正負(動作の方向性)および絶対値の大きさ(修正の程度)に基づき、事前に定義されたルールベースによってアドバイス文を動的に生成する。

数値(閾値)と具体的アドバイスの対応関係を表2に示す。本研究では、微細な誤差による誤誘導を防ぐため、各指標に許容範囲を設けている。例えばストライドであれば差分が5cm未満、肘角度であれば5度未満の場合は「現状維持」とし、過剰な指導を行わない設計とした。閾値を超えた場合、その差分の大きさに応じて指示の強度を調整する。ストライドにおいては、差分が15cm未満であれば「靴半足分」、30cm未満であれば「靴一足分」というように、選手の身体尺を用いたメタファーに変換する。また、角度情報においては「地面に突き刺す」「胸を捕手に見せる」といった、

動作感覚を想起させる表現を用いることで、数値と身体感覚のギャップを埋めることを意図している。

表 2 閾値と具体的アドバイスの対応関係

評価項目	Δ の閾値	具体的アドバイス
ストライド	$ \Delta \leq 5$	なし
	$5 < \Delta \leq 15$	靴半足分
	$15 < \Delta \leq 30$	靴一足分
	$ \Delta > 30$	一步分
前足のブロック	$ \Delta \leq 5$	なし
	$\Delta > 5$	下半身で粘る
	$\Delta < -5$	地面に突き刺す
リリースの高さ	$ \Delta \leq 5$	なし
	$5 < \Delta \leq 10$	気持ち
	$\Delta > 10$	ボール2個分
肘の伸展	$ \Delta \leq 5$	なし
	$\Delta > 5$	肘を伸ばして
	$\Delta < -5$	肘をたたみ気味
体幹の前傾	$ \Delta \leq 5$	なし
	$\Delta > 5$	胸を地面に見せる
	$\Delta < -5$	目線を上げて

生成されたアドバイスの一例を図5に示す。画面上には、5つの指標それぞれの計測値が表示され、今回特定された最重要因子については赤色の強調表示が行われる。さらに画面下部には、表2の変換に基づくアドバイス文が表示され、選手はこの画像を見るだけで「球速を上げるためには、次はどのような意識で投げればよいか」を理解することが可能となる。

フィードバック



図 5 フィードバック結果例

4. 実験と考察

本章では、提案手法の実用性および有効性を多角的に評価する。本実験を行う主たる目的は、従来の動作解析技術が抱える「専用機材による導入障壁」と「数値データの解釈困難性」という二つの課題に対し、提案システムが有効な解決策となり得るかを実証することにある。具体的には、まずシステムが設計通りに一連の処理を遂行できるかを確認する動作実験を行い、次いでLiDARと骨格推定を用いた計測手法が、投球指導において許容される精度を有しているかを検証する。さらに、実際の投手を対象とした実証実験を通じて、生成されたアドバイスが投球パフォーマンスや選手の意識変容に与える影響を定量・定性の両面から検証する。最後に、これらの実験結果に基づき、本システムの有用性と残された課題について考察する。

4.1 動作実験

本節では、実装したシステムを実際に使用し、設計通りに稼働するかを検証した。

フィードバック

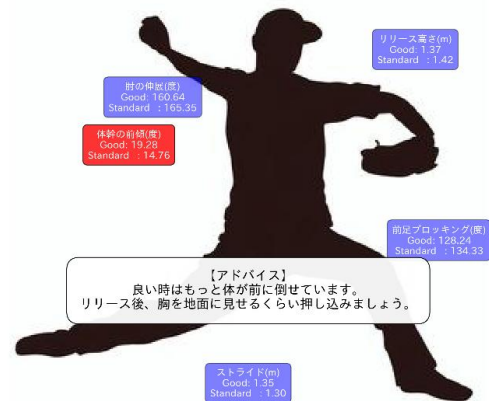


図 6 A（経験者）のフィードバック

フィードバック

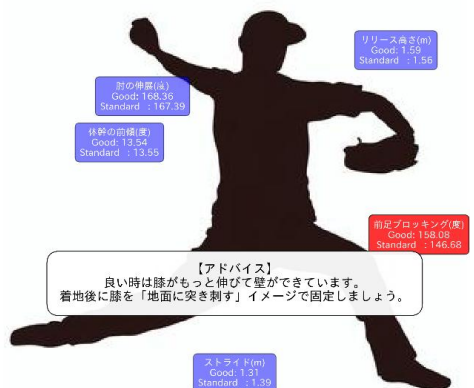


図 7 B（未経験者）のフィードバック

図 6 および図 7 の結果より、本システムは投球動作の検出から解析、フィードバックの提示までの一連のプロセスを遅滞なく実行できることが確認された。これにより、提案システムの設計および実装の妥当性が示されたといえる。

4.2 解析ポイント精度評価

本システムによって算出された解析値の信頼性を検証するため、システムによる推定値と実測値との比較評価を行った。評価項目として、「ストライド」を選定した。被験者の投球動作を本システムで解析して得られた値を「推定値」とし、同一の試行において図 8 のように足元にメジャーを設置して計測した値を「実測値」とした。これら 2 つの値の誤差を確認することで、システムの計測精度を検証した。



図 8 ストライド計測
表 3 ストライドの誤差

推定値(cm)	実測値(cm)	誤差(cm)
118.4	127.8	-9.4
141.5	139.4	2.1
133.3	120.9	12.4
137.0	140.6	-3.6
126.7	109.2	17.5



図 9 軸足が欠如した深度マップ

本システムによって算出された解析値の信頼性を検証するため、システムによる推定値と実測値との比較評価を行った。評価項目として、投球フォームの重要な指標である「ストライド」を選定し、被験者の投球動作を本システムで解析して得られた値を「推定値」、同一の試行において図のように足元にメジャーを設置して計測した値を「実測値」とした。これら 2 つの値の誤差を確認することで、システムの計測精度を検証した。本実験における評価基準としては、本システムがユーザーに提示するフィードバックの粒度が考慮されるべきである。3.6 節で述べた通り、本システムはミリ単位の厳密な動作解析ではなく、選手が直感的に修正可能な「靴半足分（約 15cm）」あるいは「靴一足分（約 30cm）」といった単位でのアドバイス生成を目的としている。したがって、計測誤差がこのフィードバック単位と同程度、あるいはそれ以下であれば、動作の改善傾向を把握するための支援ツールとして実用上の妥当性を有すると定義できる。

実験の結果、表 3 より最大で 17.5cm の誤差が生じている試行があるものの、平均誤差は約 8cm 程度であった。この誤差の主な要因としては、投球動作は高速に行われるため、モバイル端末の LiDAR スキャナによる深度取得において図 9 のように軸足のような部位の奥行情報がうまく取得できず、着地位置の特定に微細な揺らぎが発生した点が挙げられる。加えて、実測値はメジャーを用いて「靴の接地面」を基準に計測しているのに対し、システムは「足首の関節中心」を座標として検出しているという定義の差異も影響していると考えられる。しかしながら、誤差の平均は 10cm 圏内に収まっており、これはシステムが「靴半足分」単位でのアドバイスを行うための判断材料としては許容範囲内であるといえる。以上のことから、確認された誤差はシステムの仕様および環境要因によって説明可能な範囲であり、投球動作のストライド改善支援という目的において、本システムは妥当な計測精度を有していると結論付けられる。

4.3 アドバイス有効性評価

システムが提示したフィードバックおよびアドバイスが、被験者にとって有用であり、かつ実際の投球パフォーマンスに好影響を与えるかを検証した。検証は、被験者の主観に基づく「アンケート評価」と、実際の投球データに基づく「定量評価」の 2 つの側面から実施した。

4.3.1 アンケートによる定性評価

システムが提示したアドバイスの受容性を確認するため、被験者に対し、以下の 2 点についてアンケート調査を行った。

①理解しやすさ：提示されたアドバイスの意味や意図が分かりやすかったか

②実行しやすさ：アドバイスされた動作を実際の投球フォームに反映しやすかったか

回答は5段階評価(5:非常に良い ~ 1:非常に悪い)とし、アンケートの結果と被験者のコメントの一部を表4,表5に示す。

表4 被験者のアンケート結果

	理解しやすさ	実行しやすさ
A(経験者)	5	3
B(未経験者)	3	4
C(経験者)	5	5
D(未経験者)	5	4

表5 各被験者のコメント

	コメント
A 経験者	具体的な数値が表示されて理解しやすかったが、アドバイス通りに投球するのは難しかった。
B 未経験者	「壁」という表現が理解しがたかった。
C 経験者	アドバイスを意識しすぎて少し投げづらかった。
D 未経験者	アドバイスがわかりやすく実行しやすかった。

4.3.2 投球パフォーマンスの変化検証

次に、アドバイスが実際の投球結果に与える影響を検証した。システムによるフィードバックを受けた直後、各被験者に5球の投球を行わせ、アドバイスを受ける前の投球と比較して、「球速」および「アドバイスを行ったポイント」に変化が現れるかを計測した。

表6 A(経験者)のパフォーマンス変化

	通常時	好調時	フィードバック後
球速(km/h)	109.8	114.8	111.2
体幹の前傾(度)	14.8	19.3	18.6

表7 B(未経験者)のパフォーマンス変化

	通常時	好調時	フィードバック後
球速(km/h)	54.6	67.0	59.8
前足のブロックング(度)	146.7	158.1	153.4

表8 C(経験者)のパフォーマンス変化

	通常時	好調時	フィードバック後
球速(km/h)	87.4	93.7	86.9
ストライド(m)	1.43	1.58	1.61

表9 D(未経験者)のパフォーマンス変化

	通常時	好調時	フィードバック後
球速(km/h)	51.6	63.3	57.9
ストライド(m)	0.79	0.69	0.63



図10 C(未経験者)のフィードバック

投球パフォーマンスの変化検証において、被験者の熟練度による顕著な差異が確認された。

未経験者においては、表7および表9より、システムからの基本的なアドバイスが球速の向上に寄与する傾向が見られた。これは、未経験者は動作改善の伸び代が大きく、大まかなフォーム修正がパフォーマンスに直結しやすいためであると考えられる。

対照的に、表6および表8の結果より、経験者は指摘されたフォームの修正には成功したものの、それが直ちに球速向上には結びつかないケースが確認された。

特に C（未経験者）においては、アドバイス通りの動作変化は見られたものの、「いつもの感覚と異なり投げづらかった」との主観的な評価が得られている。これは、長年の経験により最適化されていた動作に対し、局所的な修正を加えたことで全体のタイミングやバランスが一時的に崩れたためであると推察される。投球動作は全身を使った一連の動きであるため、単一の部位だけを変化させても出力の向上には直結しにくい。また、アドバイスに対する適応能力や、新しい動きを自身の感覚に落とし込むまでの時間には個人差が存在することも確認された。したがって、経験者に対しては単一の指標だけでなく、例えば図 10 のようにストライドだけでなく、肘の伸展やリリースの高さにも好調時と通常時に差異が見られる。このように関連する複数のポイントを組み合わせた複合的なアドバイスを提示することで、動作の連動性を保ったまま、より効果的な指導が可能になると考えられる。

現状においても、客観的なデータに基づくフォームの見直し自体は行えており、経験者に対しては「即時的な球速向上」を保証するものではないものの、自身の感覚と客観数値のズレを認識するための「コンディション確認やフォーム矯正の支援ツール」として有効に機能したといえる。

表 4、表 5 のアンケート評価の結果も踏まえると、被験者の熟練度に応じてアドバイスの粒度や内容を動的に変更する仕組みが必要であると考察される。未経験者にはパフォーマンスに直結する基礎的な指導を、経験者にはより微細な動作解析や感覚とのすり合わせを重視した情報を提示することで、システムに対する納得感をさらに向上させることが可能である。

4.4 今後の課題：アドバイス生成システムの高度化

最後に、本研究における課題として「アドバイス生成アルゴリズムの高度化」が挙げられる。表 2 に基づく現状の固定的な閾値判定は、単一の動作特徴量を独立して評価しているため、投球フォームにおける「許容されるべき個人差」を柔軟に吸収できず、また 4.3 節で確認されたように、全身の連動性を分断して選手の違和感を招く恐れがある。投球は複数の身体部位が連動する一連の動作であるため、ある一点の修正が他部位へ及ぼす影響や、動作全体のバランスを考慮する必要がある。したがって今後は、より多くの投球データを収集し、統計的なアプローチを用いて個人差の分布をモデル化すると同時に、複数の動作特徴量間の相関関係を考慮した解析を行う必要がある。これにより、個々の投手に適応した動的な閾値設定、および動作全体の整合性を保った複合的なアドバイス生成手法の確立を目指す。

5. おわりに

深度情報と画像解析を組み合わせた本手法は、野球の投球動作に限らず他のスポーツやリハビリテーション分野への応用も期待できる。今後は、前節で述べた課題の他に、本研究では実装できなかったリアルタイム処理やモバイルデバイス単体で完結する高度な動作解析アプリケーションの実用化を目指し改良を進めていく。

参考文献

- [1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7291–7299.
- [2] D. A. Sharma, M. F. Chevidikunnan, F. R. Khan, and R. A. Gaowgzeh, "Effectiveness of knowledge of result and knowledge of performance in the learning of a skilled motor activity by healthy young adults," Journal of Physical Therapy Science, vol. 28, no. 5, pp. 1655–1659, 2016.
- [3] Record3D. <https://record3d.app/> (参照 2025-12-28).
- [4] 株式会社アクロディア. SSK TECHNICAL PITCH (テクニカルピッチ). <https://technicalpitch.net/> (参照 2025-12-28).
- [5] Ultralytics. UltralyticsYOLO. <https://github.com/ultralytics/ultralytics> (参照 2025-12-28).
- [6] OpenCV. Open-Source Computer Vision Library. <https://github.com/opencv/opencv> (参照 2025-12-28).

SQL を拡張した情報可視化言語に対する GUI 上での可換性の設計 およびライフログ分析支援の有効性評価

宇津木 萌[†] 大久保勇輝[†] 川原 一輝^{††} 富井 尚志^{†††}

[†] 横浜国立大学 大学院環境情報学府 情報環境専攻 〒240-8501 横浜市保土ヶ谷区常盤台 79-7

^{††} 横浜国立大学 理工学部 数物・電子情報系学科 〒240-8501 横浜市保土ヶ谷区常盤台 79-5

^{†††} 横浜国立大学 大学院環境情報研究院 〒240-8501 横浜市保土ヶ谷区常盤台 79-7

E-mail: [†]{utsugi-moe-yx,okubo-yuki-ph}@ynu.jp ^{††}kawahara-kazuki-cp@ynu.jp, ^{†††}tommy@ynu.ac.jp

あらまし 我々の先行研究では、PCP (Parallel Coordinates Plot) により多変量データを可視化し、その状態を SQL ライクな独自の言語表現で保存・再現可能なシステムを提案してきた。しかし、初めに提案した (PC)²DV ではデータ件数によって処理時間と描画時間が増大する問題があった。次に提案した PCPASQL + DV ではこれらの問題を解決したが、GUI 操作が不可能であった。そこで本研究では PCPASQL + DV を基盤とし、SQL を拡張した情報可視化言語と GUI の間に可換性を持つ (PC)²DV V2 (Parallel Coordinates Plot Commutative Data Visualizer Version 2) を提案する。情報可視化言語として定義を行った (PC)²L V2 の文法に対応した GUI を設計し、GUI 操作の状態を言語表現として保存・再現可能とした。本稿では、本システムを用いたライフログ分析例を挙げ、有効性を示す。また、先行研究とのパフォーマンス比較を行い、ビッグデータ分析が可能であることを示す。

キーワード 情報可視化, ユーザ支援, データ分析, 平行座標プロット

1 はじめに

近年、センサ技術の発展や各省庁等による様々なデータの公開により、実世界の状況がデータとして取得可能になった。また、ストレージの大容量化、低価格化によって取得したデータの全てを蓄積・保存することが可能になった。これにより、人々の日常的な生活や活動を記録した「ライフログ」をデータとして容易に取得し、利用できるようになった。ライフログには個人ごと場所ごとの特徴が含まれており、そのようなデータ固有の知見を示すことは、個々の事例においても社会全体においても有用である。また、今後の社会では、様々なソースから多様なデータが取得できるようになると考えられ、日々のデータを蓄積するライフログは膨大かつ多様な多変量データとなる。このような、出処や形式、粒度、種類が異なるデータを結合して分析を行うことは重要である。以上のことから、関係データベース (RDB: Relational Database) および SQL を用いた分析が有効であるといえる。しかし、SQL による RDB の操作結果は表形式であり、その形式のまま知見を得ることは容易ではない。また、大量のデータから有用な知見を得るためには、操作を繰り返し行う必要がある。そのため、データを可視化しながら、試行錯誤を伴う分析をすることが必要であると考えられる。

そこで、我々の先行研究 [1-4] では、平行座標プロット (PCP: Parallel Coordinates Plot) [5,6] を用いたデータ可視化システムを提案してきた。PCP ではデータの属性を平行な軸に割り当て、軸を結ぶ折れ線 1 本でデータ 1 件を表す。PCP でデータ全件を可視化することにより、大まかな属性間の相関やクラスタ、外れ値を容易に把握することができる。

PCP で可視化するデータは表形式であるため、PCP 上で SQL と同等の表に対する操作が可能である。この特徴に注目し、我々は複数の属性からなるデータを PCP により可視化し、その状態を SQL ライクな言語表現により保存・再現することが可能なシステム (PC)²DV (Parallel Coordinates Plot Commutative Data Visualizer) を提案してきた [1-3]。 (PC)²DV は GUI 上での選択・射影・結合といったリレーショナル代数演算によるデータ操作が可能である。また、分析の過程において、分析者はデータ操作やデータ可視化の状態を、SQL ライクな独自の言語表現 (PC)²L (Parallel Coordinates Plot Commutative Language) により保存・再現することが可能である。

しかし、(PC)²DV では件数が膨大なデータを扱う場合、データの処理時間と描画時間が長いという 2 つの問題があった。その原因として、(PC)²L ではデータを取得するための SQL 部分と、取得したデータの PCP 表示方法を記述する可視化部分が混在していたことが挙げられる。(PC)²DV は最初にデータを全て取得し、SQL に該当する部分をクライアントのフロントエンドで独自に処理を行っていたため、データ処理が最適化されておらず、時間がかかっていた。また、PCP の線の描画に、描画ライブラリである D3.js が用いられていたが、D3.js では CPU で線の描画を行うため、GPU を活用できていなかった。そのため、PCP の描画時間が長くなっていた。そこで先行研究 [4] では、これらの問題を解決した新たなデータ可視化ツールである PCPASQL + DV (Parallel Coordinates Plot Augmented SQL + Data Visualizer) を提案した。PCPASQL + DV では 2 つの改善を行った。1 つ目として、新たに情報可視化言語である PCPASQL (Parallel Coordinates Plot Augmented SQL) を定義した。PCPASQL では、SQL 部分と可視化部分を分離

し、SQL 部分は DBMS に処理させることで処理速度を向上させた。2 つ目の改善として、実装に OpenGL を使用したことがあげられる。それにより、PCPASQL + DV では、OpenGL による GPU アクセラレーションを用いることで描画速度を向上させた。

PCPASQL + DV はこれらの改善に注力したため、 $(PC)^2DV$ では可能であった GUI 上での操作が実装されておらず、PCPASQL のクエリ文入力でのみ操作が可能であった。そのため、表示された PCP を用いた直観的な操作ができず、試行錯誤を伴う分析には適していないと考えられる。そこで、本研究では PCPASQL + DV を基盤とし、GUI による PCP の操作を追加した $(PC)^2DV V2$ (Parallel Coordinates Plot Commutative Data Visualizer Version 2) を提案する。 $(PC)^2DV V2$ は、GUI 上で直感的な操作を可能とし、操作結果を独自の情報可視化言語である $(PC)^2LV2$ (Parallel Coordinates Plot Commutative Language Version 2) で保存・再現ができるデータ分析支援ツールである。GUI の操作手順はリレーショナル代数を参考に設計し、基本演算に対応したものとす。これにより、GUI による操作で RDB に対する操作を広範でカバーすることが可能になり、SQL を拡張した言語である $(PC)^2LV2$ で操作過程を表現できるようになる。また、ライフログデータ分析の例として、建物の需要電力データの分析を行う。これにより、試行錯誤を伴うライフログデータ分析における $(PC)^2DV V2$ の有用性を示す。

2 関連研究

2.1 データ可視化とデータ分析

PCP は 1985 年、Inselberg によって初めて概念が定義された [5]。それ以降、PCP を用いたデータ分析の研究が盛んに行われている。Johansson らによれば、PCP の研究カテゴリーは次の 4 つに分類される [6]：

- (1) PCP の (属性) 軸レイアウト
- (2) PCP の Clutter 軽減方法
- (3) PCP の実応用例の提示
- (4) PCP と他のデータ解析手法との比較

しかし、PCP の見せ方についての議論がほとんどであり、PCP の操作過程に着目した議論はされていない。また、操作の過程でデータ自体にリレーショナル代数のような演算を加えながら操作するものは議論されていない。

複数の属性からなるデータを可視化するその他の一般的な手法として、複数の散布図を表示する散布図行列が挙げられる [7]。散布図行列は、2 つの属性間の相関を直感的に把握できるが、散布図数が属性数の 2 乗に比例して増加する。そのため、分析過程でデータに対し結合の操作を行うことには不向きであるといえる。また、Bouali らは、対話型遺伝的アルゴリズムにより可視化手法の推薦を行い、データや利用者の要求に応じてより適切な可視化手法の選択を支援するシステムを提案した [8]。我々が提案してきた $(PC)^2DV$ 、PCPASQL + DV と、本稿で提案する $(PC)^2DV V2$ は関係代数演算における選択・射影・結

合が表現可能な可視化システムであるため、データ一件を 1 本の折れ線で表し、詳細に参照・分析可能である PCP が適切である。データを可視化し、分析を行う研究 (Visual Analytics) が盛んに行われている。Cui [9] による分類では、PCP を用いたデータ分析および本研究は多次元データをアルゴリズムに基づき変形し二次元空間で可視化する "Multi-Dimensional-Transformation-2D" かつ、データ操作によりデータや可視化空間を探索する "Exploratory-Oriented" に分類される。

Visual Analytics の中で、我々の手法と同様にインタラクティブな操作と PCP による可視化を組み合わせた可視化・分析手法の提案がされている。Itoh らは、属性軸間の相関に基づいてインタラクティブに次元削減を行い、PCP から所望する情報の発見を支援するシステムを構築した [10]。Zhou らは、エントロピーの概念を導入することで、PCP の属性軸の整列順序をクラスタに基づいて決定する手法を提案した [11]。Bok らは、任意の属性値を基準としたデータの分布を表すヒストグラム (PHP : Parallel Histogram Plot) を PCP 上に表示し、PCP の軸上のデータの分布や属性間の相関の把握を支援する手法を提案した [12]。Gruendl らは、時間を新たな次元と考え、PCP の二軸間の奥行き方向に時間軸を導入し、PCP と時系列プロットを統合した、時間依存のデータを可視化・分析する手法を提案した [13]。Cibulski らは、相互関係のあるデータセットの PCP を結合した複合平行座標により、複数のコンポーネントを同時に分析する手法を提案した [14]。これらの研究と比較して我々は、PCP 上でのデータ操作が関係代数演算と同等であり、可視化の状態と言語が可換である点に着目している。

2.2 データ操作過程管理 (Data Provenance)

データやシステムの操作過程を管理する研究 (Provenance) が行われている [15]。Herschel らは文献 [15] 内で、特にデータやシステム、プログラミングコードなどの操作過程や操作の意図を保存することは、複雑なデータ処理を支援するために重要なことであると述べている。さらに、分析結果データの操作過程や操作の意図を示すことは、SQL のような関係代数演算をサポートする問合せ言語で記述することが有効であるとも述べている。この点において、 $(PC)^2LV2$ を用いて $(PC)^2DV V2$ のデータの操作過程の状態を保存することは有効な手段であるといえる。

また、データやシステムの操作過程を保存することでユーザの支援を行う手法が提案されている。Waldner らは、PC のアプリケーションの閲覧履歴や操作履歴を保存し、それらを時系列が理解できるように可視化することで、ユーザが過去に行った情報探索の詳細を再現する支援を行った [16]。Mindek らは、画像データと、分析過程で利用する他のソースのデータを同時に表示し、分析者の文脈を含蓄したスナップショットを保存することで、シミュレーションデータの可視化や文書分析の支援を行った [17]。Psallidas らは、インタラクティブ性を持つ可視化ツールの操作過程を宣言的操作クエリに変換することによる有用性を述べた [18]。Gratzl らは、PCP やヒートマップ、散布図行列など様々な可視化手法を組み合わせることで複数のソースか

ら得られたデータとその解析過程を可視化し、データ解析の支援を行った [19].

これらの手法と比較して我々の手法は、「可視化システムのデータ解析過程を可視化して見せる」のではなく、「SQL に類似した言語を用いてデータ分析の過程の任意の状態を保存し、問合せ言語として一般的な SQL を熟知するデータ分析者を支援する」ものであり、立場が異なる。また、言語を用いて状態を保存することにより、言語の一部を書き換えるだけで容易にデータ分析の改善をすることができる。その点でこれらの研究と比較して優位性をもつ。

また、Holger らは分析プロセスの可視化状態と操作を、検索可能なグラフ構造として保存・再現する分析支援システムを提案した [20]. 彼らの手法は、分析における過去の状態を言語情報により保存・再現可能である、という点が我々と共通する。しかし、彼らは「グラフ構造により分析の履歴を示し」、操作を含めた分析過程を「検索可能な形ですべて保存する」一方で、我々の手法は「可視化状態と言語が可換である」という点を重視し、分析過程において GUI 上で構成された可視化結果の状態をクエリ言語の形式でことにより支援を行う。この点において、我々の研究とは立場が異なる。

3 (PC)²DV と PCPASQL + DV における問題

我々は先行研究 [1-3] において PCP によるデータ可視化システムである (PC)²DV を提案してきた。さらに、その後の先行研究 [4] では (PC)²DV の問題点を改善した PCPASQL + DV を提案した。これらの先行研究と本稿で提案する (PC)²DV V2 をまとめた表を表 1 に示す。本章では (PC)²DV と PCPASQL + DV の概要と問題について説明する。

3.1 (PC)²DV の概要

(PC)²DV では、PCP に対し GUI 上でインタラクションを行い、SQL ライクな独自の言語 (PC)²L によりデータ操作を行った状態を保存・再現することが可能である。(PC)²DV は Web ブラウザを通して多くの端末から利用できるように構築した。このシステムの操作の流れを以下に示す。

- (1) 任意のデータソースへ接続し、データを読み込む。
- (2) リレーションを PCP により可視化する。
- (3) PCP に対して GUI 上でインタラクション（データ操作）を行う。この際、データ操作結果はリアルタイムに反映される。
- (4) データ分析者が任意に、(2)、(3) の分析過程の状態を (PC)²L で保存する。
- (5) (2) から (4) を繰り返す。その際、過去の状態に戻りたい場合は該当する (PC)²L を入力し、その状態を再現する。
- (6) データ分析者が所望の可視化結果を獲得する。

(PC)²DV には 2 つの問題が存在した。それは、データ処理時間が長いこととデータ描画時間が長いことである。

1 つ目の問題として、分析対象データの処理に時間がかかっ

てしまうことがあげられる。(PC)²DV では GUI 上での操作や、(PC)²L の入力結果のデータを作成する処理をフロントエンドで独自に行っていた。(PC)²DV の分析対象となる多変量データは JavaScript の連想配列となり、インデックスを利用したデータ構造にはなっていない。また、クエリの最適化処理も行われていない。そのため、データ件数が増えるにつれて、データ操作処理の時間は線形時間で増大してしまう。

2 つ目の問題は、データ処理を行った後の PCP が描画されるまでの時間が長いことである。(PC)²DV ではデータ描画の API として D3.js が使われていた。D3.js は主に SVG を用いて描画を行う。また、DOM ツリーを介してブラウザ上に要素をレンダリングするため、処理は主に CPU に依存する。その結果、描画する線や要素の数が増えるとデータ描画時間が長くなる問題がある。これは、データ件数が膨大になるライブログデータの分析において支障をきたすと考えられる。

3.2 PCPASQL + DV の概要

PCPASQL + DV は、(PC)²DV のデータ処理時間とデータ描画時間の問題を改善した可視化システムである。このシステムは、独自の情報可視化言語である PCPASQL を用いて、データを PCP として可視化し、分析を行うことができる。以下のような手順で分析を行う。

(1) データ分析者が事前に PCPASQL を指定のファイルに記述する。

(2) 指定された PCPASQL に従って、PCPASQL+DV が PCP を描画する。

PCPASQL + DV では、(PC)²DV で問題となっていたデータ処理時間とデータ描画時間に対して 2 つの改善を行った。1 つ目として情報可視化言語である PCPASQL を提案し、SQL 部分（データ処理）と可視化部分（描画設定）を明確に分けた。SQL 部分と可視化部分に分けることで、SQL 部分の処理を全て DBMS に任せることが可能になり、以下の機能を活用できるようになった。これにより、データ処理速度を向上させることが可能となった。

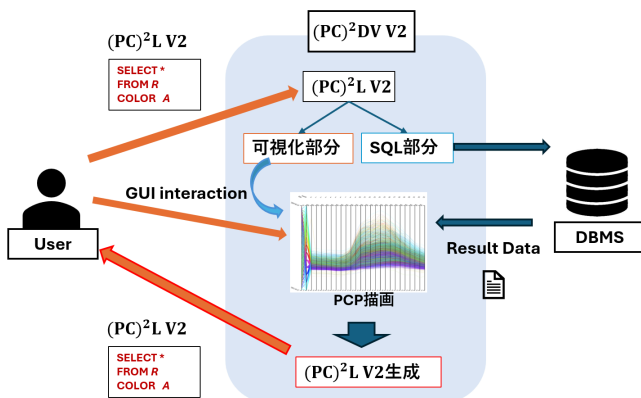
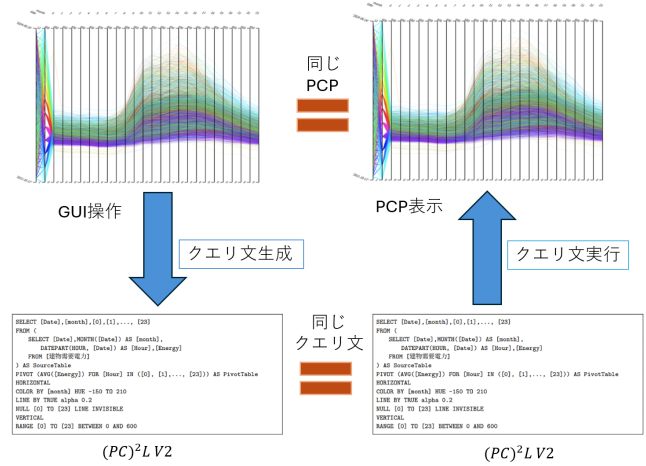
- インデックス構造の活用
- 実行プランの最適化

2 つ目の改善として実装に OpenGL を用いた。OpenGL とはグラフィックを描画するための API であり、C,C++などの言語で利用可能である。OpenGL による GPU アクセラレーションの結果として PCP の線の描画速度を向上させた。

PCPASQL + DV では、ビッグデータに利用可能なデータ分析システムを実装することを優先した。そのため、PCP と情報可視化言語 PCPASQL の間に可換性を持たせることを諦めた。また、GUI 上での操作機能は実装されておらず、PCPASQL のクエリ文を入力することでのみデータ操作を実行可能であった。つまり、(PC)²DV の特徴である、リアルタイムにデータ操作が反映される GUI と、その状態と等価である論理的な言語表現を相互に活用した分析が不可能であった。

表 1 先行研究の比較

可視化システム	可視化言語	可換性	GUI 上での操作	使用ライブラリ	文献
(PC) ² DV	(PC) ² L	○	○	D3.js	[1-3]
PCPASQL + DV	PCPASQL	×	×	OpenGL	[4]
(PC) ² DV V2	(PC) ² L V2	○	○	OpenGL	本稿

図 1 (PC)²DV V2 の概要図 2 GUI と (PC)²L V2 の可換性の概要

4 (PC)²DV V2 の概要と設計

本章では、新たに提案するデータ可視化システムである (PC)²DV V2 の概要と設計について述べる。

4.1 (PC)²DV V2 の概要

(PC)²DV V2 の概要を図 1 に示す。(PC)²DV V2 では、先行研究 [4] で提案した PCPASQL + DV の機能に追加し、PCP に対し GUI 上でインタラクションを行いつつ、その操作結果を言語によって保存・再現することを可能にする。本システムでは、新たに情報可視化言語 (PC)²L V2 を定義し、GUI と (PC)²L V2 が可換性をもつように設計する。想定する可換性の概要を図 2 に示す。これにより、ライフログのような膨大な件数を持つデータの分析に耐えうる、直感的な操作で試行錯誤が可能なデータ可視化システムを実現する。(PC)²DV V2 では、(PC)²DV と同様に、以下のようなデータ操作手順により分析を行う分析者に対して支援を行うことを想定する。

- (1) 任意のデータソースへ接続し、データを読み込む。
- (2) リレーションを PCP により可視化する。
- (3) PCP に対して GUI 上でインタラクション（データ操作）を行う。
- (4) データ分析者が任意に、(2)、(3) の分析過程の状態を (PC)²L V2 で保存する。
- (5) (2) から (4) を繰り返す。その際、過去の状態に戻りたい場合は該当する (PC)²L V2 を入力し、その状態を再現する。
- (6) データ分析者が所望の可視化結果を獲得する。

4.2 PCP に対する SQL 操作

PCP ではリレーションの可視化を行うため、SQL のリレー

ショナル代数演算の操作が可能である。本研究では、リレーショナル代数演算である選択演算、射影演算、直積演算の 3 つを PCP 上での SQL 操作として考える。リレーショナル代数演算については、文献 [21] を参考にした。本節では、これらの演算と PCP 上での操作の関係を述べる。ここで用いる記号の定義を表 2 に示す。

選択演算 リレーションにおける選択演算は、属性値に基づいてタプルを抽出する操作である。これは、PCP 上に表示された折れ線のうち、条件を満たすものを選択的に表示する操作に対応する。SQL では WHERE 句に相当し、属性 A_i に対して行われる操作は「WHERE $p(A_i)$ 」の形式で記述される。ここで p は「 $A_i \theta v$ 」、「 A_i BETWEEN v_1 AND v_2 」、「 A_i LIKE s 」などの条件式である。

射影演算 射影演算は、リレーション $R(A_1, A_2, \dots, A_n)$ から、 $R[A_{i_1}, A_{i_2}, \dots, A_{i_k}]$ のように属性を切り出す操作である。PCP 上では軸を切り出す操作に対応する。SQL では SELECT 句による属性の列挙に相当し、列挙された順序に従って、PCP の軸が左から右へ配置されるものとする。

直積演算 直積演算は、2 つのリレーション $R(A_1, A_2, \dots, A_n)$ と $S(B_1, B_2, \dots, B_m)$ から、すべてのタプルの組合せを生成し、新たなリレーション $R \times S(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$

表 2 記号の定義

記号	概要
A_i	属性
$R(A_1, A_2, \dots, A_n)$	リレーション
$R[A_i \theta A_j]$	R の A_i と A_j 上の θ -選択演算
$R[A_{i_1}, A_{i_2}, \dots, A_{i_k}]$	R の $A_{i_1}, A_{i_2}, \dots, A_{i_k}$ 上の射影演算
$(1 \leq i_1 < i_2 < \dots < i_k \leq n)$	

を構成する操作である。PCP 上では、2つのリレーションに対応する折れ線集合を組み合わせ、すべての組合せとして表示する操作に対応する。SQL では FROM 句における複数テーブルの列挙に相当し、その後 WHERE 句で条件を付与することにより結合演算として実現される。

4.3 PCP 可視化の独自操作

本節では、PCP 可視化に対する独自操作について述べる。PCP に対する操作は水平方向の折れ線に対するものと、垂直方向の軸に対するものに分けられる。

4.3.1 折れ線に対する操作

折れ線に対する操作は、「線の色付け」、「線の太さや種類の変更」、「NULL 値の扱いの変更」の3つに分けられる。

「線の色付け」は、任意の属性 A の値を基準として折れ線の色分けを行う操作である。「線の太さや種類の変更」では、色分けを行う場合と同様に、任意の属性 A の値を基準として折れ線の太さ、透明度、線種などを変更する。「NULL 値の扱いの変更」は、NULL 値を含むデータを PCP の折れ線としてどのように描画するかを決定する操作である。

4.3.2 軸に対する操作

軸に対する操作は、「軸の色付け」、「軸の太さや種類の変更」、「軸の範囲変更」の3つに分けられる。

「軸の色付け」は、折れ線に対する操作と同様に、任意の属性 A の値を基準として軸の色を変更する操作である。「軸の太さや種類の変更」も、折れ線に対する操作と同様に、任意の属性 A の値を基準として軸の太さ、透明度、線種などを変更する。「軸の範囲変更」は、軸の最大値および最小値を指定して表示範囲を変更する操作である。軸の範囲を狭めることで、折れ線の傾向をより詳細に可視化することが可能となる。

4.4 情報可視化言語 (PC)²LV₂

本節では、(PC)²DV V₂ で使用する情報可視化言語である (PC)²LV₂ について述べる。PCP のインタラクションの過程を保存・再現するためには、PCP の GUI 上での各操作と対応する言語の定義が必要である。そこで、本稿では情報可視化言語 (PC)²LV₂ を提案する。(PC)²LV₂ は、PCP におけるリレーショナル代数表現と対応し PCP における操作結果を言語化して保存する機能と、言語をもとに PCP を描画する機能を有する。また、PCPASQL + DV と同様に SQL 部分と可視化部分が分かれており、SQL 部分を分離して DBMS に送信することでデータ処理の高速化が可能である。

本言語の SQL 部分では、リレーショナル代数の基本演算が表現可能であるものとする。基本演算として、4.2 節で述べたように、選択演算、射影演算、結合演算をそれぞれ SQL 部分の WHERE 句、SELECT 句、FROM 句として表現し、PCP におけるインタラクションと対応するように定義する。以下では、これら基本演算と PCP の関係について述べる。

- **WHERE 句**：選択演算。軸に対して条件や範囲選択することで、表示する PCP の線 (タプル) を絞り込む。
- **SELECT 句**：射影演算。テーブルのカラムから PCP

で表示する軸を指定し、任意の順番に並べる。

- **FROM 句**：直積演算。複数のテーブルを読み込んだ場合、それぞれが持つ線 (タプル) をつないで軸の追加を行う。これらに加え、可視化部分として、PCP の描画方法を記述する句である HORIZONTAL 句と VERTICAL 句も利用可能とする。HORIZONTAL 句と VERTICAL 句の仕様に関して、以下に示す。

- **HORIZONTAL 句**

- **COLOR 句**：「COLOR A_i WITH」で軸を指定し、軸の値を基準にして線を色付ける。色付けの方法はグラデーションと条件式を記述する2つがある。グラデーションでは、「HUE」(色相)、「SATURATION」(彩度)、「VALUE」(明度)、「ALPHA」(透明度)のどれかを選択し、最小値・最大値を記述する。条件式では、その後ろに「赤」、「緑」、「青」を記述することも可能である。

- **FIX LINE 句**：線の太さや透明度、線種の変更を記述する。「FIX LINE IN [条件] TO」とすることで、変更する対象となる PCP の線を指定する。線の太さを変更する場合は「WIDTH [太さ]」、透明度は「ALPHA [透明度]」、線種を点線にする場合は「DASHED」で記述する。

- **FIX NULL 句**：リレーションの NULL を PCP の線でどのように描画するか記述する。「FIX NULL IN A_i TO」で対象の軸の指定を行う。描画方法は「TOP」、「BOTTOM」、「INTERPOLATE」、「VALUE INVISIBLE」、「LINE INVISIBLE」の5つである。

- **VERTICAL 句**

- **COLOR 句**：「COLOR A_i 」で色付けを行う軸を指定する。

- **FIX LINE 句**：軸の太さや透明度、線種の変更を記述する。「FIX LINE IN A_i TO」とすることで、変更する対象となる軸を指定する。太さを変更する場合は「WIDTH [太さ]」、透明度は「ALPHA [透明度]」、線種を点線にする場合は「DASHED」、非表示にする場合は「INVISIBLE」で記述する。

- **SCALE 句**：軸の範囲を記述する。「SCALE [軸] BETWEEN [最小値] AND [最大値]」とすることで、軸の範囲を最小値から最大値で描画することが可能である。対象となる軸は「 A_i 」と1本のみではなく、「 A_{i_1} TO A_{i_2} 」のように2本記述することで、この2本の間に存在する軸も含めて範囲を指定することができる。

4.5 PIVOT 操作を用いた時系列データ分析

PIVOT とは SQL において、リレーション名の行を削減し列に変換する操作である。この操作により、データを束ね件数を減らしつつ属性数を増やすことでデータ一件の単位を変えることが可能である。たとえば、データ1件が1時間を表すリレーションを、データ1件で1日を表す周期データへ変換することができる。

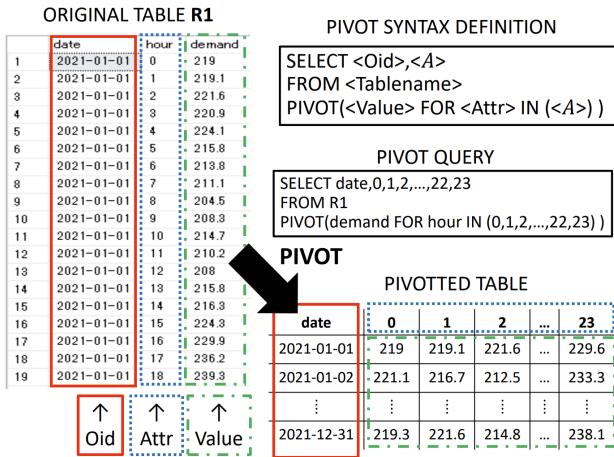


図 3 PIVOT 操作の概要

PIVOT 操作を実行するにあたり、必要な要素として < Oid >、< Attr >、< Value >がある。< Oid >とは、PIVOT 後のテーブルの主キーとなる属性であり、< Oid >の値ごとにデータが束ねられる。< Attr >とは、PIVOT 後のテーブルのカラム名となる値を持つ属性である。< Value >とは、PIVOT 後のテーブルの各カラムの値となる値を持つ属性である。

図 3 では属性 [date]、[hour]、[demand] からなるテーブル R1 に対して PIVOT の操作を行う。< Oid >となる属性は [date] であり、PIVOT 後のテーブルはデータが日ごとに束ねられ、レコード 1 件が日ごとのデータとなる。< Attr >となる属性は [hour] であり、この値が PIVOT 後のテーブルのカラムとして並ぶ。< Value >となる属性は [demand] である。このようにして [date] ごとに [hour] ごとの [demand] の値が PIVOT のテーブルの値として並ぶ。

我々が分析対象とするライフログは、時系列データの形式として蓄積される。そのため、PIVOT 操作を行うことで、時系列データを周期データに変換し、PCP で折れ線グラフのような分析が可能になる。

5 (PC)²DV V2 を用いたライフログ分析

5.1 分析の目的

本章では、4 章で述べた (PC)²DV V2 を用いて建物の需要電力データを分析する。需要電力データは、1 時間ごとに取得されている。このデータを PIVOT 操作により、1 日ごとの周期データに変換し PCP で日々の電力波形を可視化する。これにより、需要電力の季節ごとの違いや、時刻による傾向を可視化する。次に、曜日のデータと結合し、休日と平日の需要電力の違いを可視化する。また、(PC)²L V2 により保存したデータ操作結果を利用して、複数の建物の需要電力の傾向を比較する。以上について PCP を操作して説明することで、ライフログデータ分析における (PC)²DV V2 の有用性を示す。

5.2 使用データと分析方法

本稿では、分析対象の建物の需要電力データとして、横浜国立大学の総合研究棟（建物 A）の 1 時間ごとの需要電力の実

表 3 リレーション「建物需要電力_建物 A」の属性

属性名	説明
month	月 (mm)
day	日付 (dd)
hour	時間 (0~23)
demand	1 時間ごとの建物の需要電力量 (kWh/h)

表 4 PIVOT 操作後のリレーション「建物需要電力_建物 A」の属性

属性名	説明
month	月 (mm)
day	日付 (dd)
0~23	1 時間ごとの建物の需要電力量 (kWh/h)

表 5 リレーション「曜日」の属性

属性名	説明
month	月 (mm)
day	日付 (dd)
weekday	曜日

データ¹を使用する。データ取得期間は 2019 年 1 月 1 日から 2019 年 12 月 31 日までであり、データ件数は 24 時間 × 365 日で 8760 件である。このリレーション「建物需要電力_建物 A」の属性について表 3 に示す。なお、データは Microsoft SQL Server 上のデータベースに格納されている。このデータベースに ODBC 接続し、データを取得する。

5.1 節で述べたように、取得されたデータは時系列データになっており、そのまま PCP で可視化しても知見を得ることは難しい。そこで、日付 ([date]) を主キーとして、時間 ([hour]) がカラム名となるように PIVOT 操作による変換を行う。変換を行い周期データとなったリレーション「建物需要電力_建物 A」を表 4 に示す。

このリレーション [建物需要電力_建物 A] について、(PC)²DV V2 を用いて以下のプロセスで可視化・分析を行う。結合するリレーション「曜日」を表 5 に示す。

- (1) データソースに接続してデータを取得し、PCP で可視化する。
- (2) 可視化されたデータに対して、PIVOT の操作を行い、日ごとの周期データに変換する。
- (3) 変換されたデータを PCP で可視化する。
- (4) 同一の尺度で値を比較するため、軸の範囲を統一する。
- (5) 季節ごとの傾向を見るため、月を基準に折れ線の色付けする。
- (6) リレーション「曜日」と結合する。
- (7) 結合した曜日を基準に折れ線の色付けする。

5.3 需要電力データの分析

まず、1 つの建物の需要電力データ「建物需要電力_建物 A」について (PC)²DV V2 上での操作・分析を行う。5.2 節で述

1: 横浜国立大学施設部, <http://shisetsu.ynu.ac.jp/gakugai/shisetsu/> (学内限定アクセス)

べたように、リレーション「建物需要電力_建物 A」はそのまま可視化しても周期的な情報を読み取るのは難しい。そこで、PIVOT 操作による変換を行い、1日ごとの周期データとして可視化する。変換後の PCP とその状態を表す (PC)²LV2 によるクエリ文を図 4 に示す。図 4 では、周期データとして可視化できているが、[0] から [23] までの各軸の範囲にばらつきがあるため、軸の範囲を指定して揃える操作を行う。軸の範囲を 0kWh/h から 500kWh/h に揃えた PCP を図 5 に示す。これにより、1日の電力推移を波形として把握することができる。この図から、この建物の需要電力は人々の活動に合わせて7時ごろから増加し、昼過ぎから夕方間に大きくなり、夜になるにつれて減少するという傾向があることがわかる。さらに、折れ線が上下方向に分布していることから、日によって需要電力に違いがあることがわかる。

ここで、軸 [month] を基準にして折れ線に色を付ける操作を行う。操作後の PCP は図 6 のようになり、季節ごとの大きな違いを把握することができる。この図から、夏(緑)や冬(青)の昼間の需要電力は大きく増加することがわかる。また、春(赤)の需要電力は他の季節と比べて低いことがわかる。さらに、春(赤)の折れ線は、昼間に大きく増加するクラスタと、増加が比較的小さいクラスタに分かれていることが確認できる。

この2つのクラスタに分かれる原因として、休日と平日の違いが関係していると考え、その違いを明確化するために、リレーション「曜日」の結合を行う。結合を行うために、リレーション「建物需要電力_建物 A」に追加して、リレーション「曜日」のデータを取得し可視化する。ただし、単に可視化を行った状態では直積演算をとっていることになり、すべてのタプルの組合せて繋がっている状態である。そのため、選択演算で日付が一致している折れ線のみを抽出する操作を行う。この操作により、曜日ごとの需要電力の傾向を可視化することができる。土曜日と日曜日を青色、月曜日から金曜日を赤色とした PCP を図 7 に示す。この図を見ると、土日を表す青色の線が下部に集中し、平日を表す赤色が上に広がっていることが確認できる。よって、休日は需要電力が低くなり、反対に平日は高くなることがわかり、春の需要電力が上下2つのクラスタに分かれている原因が、休日と平日の違いであると確認できる。このように、データ操作を行いながら PCP で描画することで、季節ごとの差異や休日と平日の違いを可視化することが可能である。

5.4 (PC)²LV2 による操作結果の利用例

(PC)²DV V2 上での PCP の操作後の状態は言語 (PC)²LV2 のクエリ文で保存可能である。保存したクエリ文は、同じ構造の異なるリレーションに対して使用することができる。5.3 節では、総合研究棟(建物 A)の需要電力について分析を行ったが、この操作結果を保存し、別の建物の需要電力のデータに適用することが可能である。

本節では、図 6 の状態を保存した (PC)²LV2 のクエリ文を利用して、別の建物の需要電力データを可視化し、比較を行う。比較する建物は、横浜国立大学の環境情報 1 号棟(建物 B)と、経済・経営系(建物 C)である。これらの建物の需要電力デー

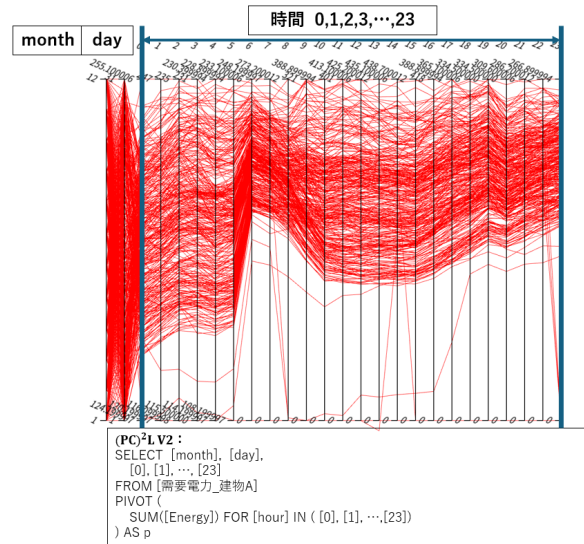


図 4 「建物需要電力_建物 A」を PIVOT で変換したの PCP

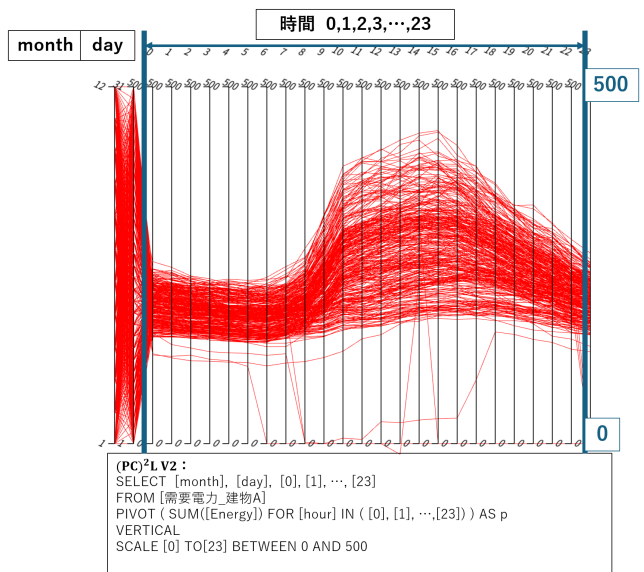


図 5 「建物需要電力_建物 A」の軸を揃えた PCP

タのリレーションは、「建物需要電力_建物 A」と同じ構造をしている。

まず、保存したクエリ文にある FROM 句のリレーション名を、環境情報 1 号棟(建物 B)のデータである「建物需要電力_建物 B」に変更する。変更したクエリ文を図 8 に示す。このクエリ文を (PC)²DV V2 で実行することで、図 9 の PCP が描画される。このように、(PC)²LV2 により 5.3 節で行った操作を再び行うことなく、クエリ文を実行するだけで、操作の再現が可能である。この図を図 6 と比較することで、建物ごとの需要電力の違いを把握できる。さらに、経済・経営系(建物 C)のリレーション「建物需要電力_建物 C」も PCP で可視化する。「建物需要電力_建物 B」の分析と同様に、保存したクエリ文のリレーション名を「建物需要電力_建物 C」に変更し、(PC)²DV V2 で実行する。これにより描画された PCP を図 10 に示す。

ここで、建物 A の図 6 の PCP と、建物 B の図 9 を比較す

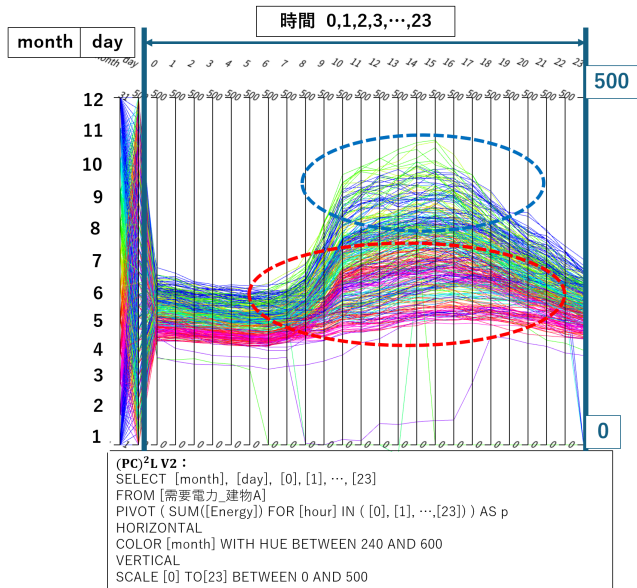


図 6 「建物需要電力_建物 A」の月を基準に色分けした PCP

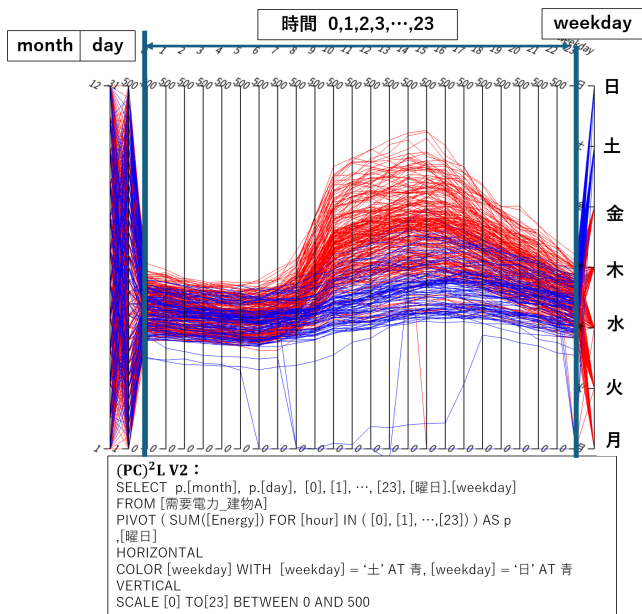


図 7 「建物需要電力_建物 A」と「曜日」を結合した PCP

```
SELECT [month], [day], [0], [1], ..., [23]
FROM [需要電力_建物B]
PIVOT ( SUM([Energy]) FOR [hour] IN ( [0], [1], ..., [23]) ) AS p
HORIZONTAL
COLOR [month] WITH HUE BETWEEN 240 AND 600
VERTICAL
SCALE [0] TO[23] BETWEEN 0 AND 500
```

図 8 リレーション名を変更した (PC)²LV2 クエリ文

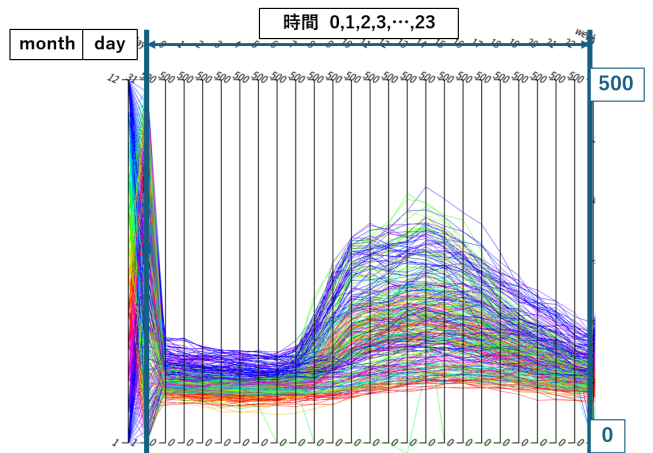


図 9 「建物需要電力_建物 B」の PCP

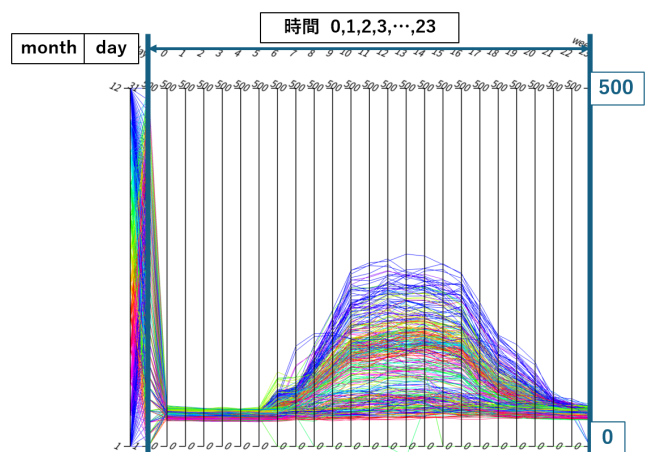


図 10 「建物需要電力_建物 C」の PCP

ると、おおまかな傾向は同じであることがわかる。しかし、夜間の需要電力に差がある。この理由として、建物 A は夜間も稼働し続けているサーバや実験機器などの数が多いことが考えられる。また、建物 A の図 6 の PCP と、建物 C の図 10 を見比べると、傾向が異なっているのがわかる。建物 C の夜間における需要電力は日ごとの差異が小さく、ほぼ一定であることが確認できる。また、夕方から夜間にかけて需要電力が急激に減少している様子が把握できる。これは、建物 C では夜間に稼働している機器がほとんど存在せず、夜遅くまで滞在する人が少ないことが要因であると考えられる。本稿では、(PC)²DV V2 を用いて 3 つの建物需要電力データの分析を行ったが、リレーションの構造が同一であれば、他の建物データに対しても同様の分析が可能である。

6 パフォーマンス評価

本章では、(PC)²DV V2 と (PC)²DV, PCPASQL + DV のパフォーマンスの比較を行い、(PC)²DV V2 によるビッグデータ分析が可能であることを示す。

6.1 (PC)²DV V2 におけるパフォーマンスの定義

(PC)²DV V2 のパフォーマンスには、SQL を実行して DBMS からデータを取得するクエリ処理時間と、取得したデータを PCP として描画するデータ描画時間の 2 つがある。ここで、ユーザの操作終了から PCP で描画するデータセットを作成するまでをクエリ処理時間、画像処理を開始してから、画像処理が終了するまでの時間をデータ描画時間と定義した。クエリ処理時間とデータ描画時間を合わせたものが合計時間となる。

表 6 クライアント PC の実験環境

項目	説明
OS	Windows 11 Education
CPU	Intel(R) Core(TM) i5-13400
メモリ	32GB
GPU	Intel UHD Graphics 730

表 7 サーバ PC の実験環境

項目	説明
OS	Windows Server 2022
DBMS	Microsoft SQLServer 2022 developer
CPU	Intel(R) Core(TM) i7-13700 2.10GHz
メモリ	128GB
ディスク	WD BLACK SN850 HS 2TB × 3
ネットワーク転送速度	2.5Gbps

表 8 実験用テーブルの属性

属性	インデックス	データ型	値の範囲
id_index	あり (クラスター化)	int	1-10,000,000
int_data_index	あり (非クラスター化)	int	1-30
float_data	なし	float	1-10

```
SELECT [id_index],[int_data_index],[float_data]
FROM [TableName]
WHERE [id_index] <= N
```

図 11 実験で用いたクエリ

6.2 評価概要

本実験ではクライアント PC で (PC)²DV V2, (PC)²DV, PCPASQL + DV を実行して、サーバ PC から実験用テーブルのデータを取得して実験を行う。クライアント PC の環境を表 6, サーバ PC の環境を表 7 に示す

サーバ PC の Microsoft SQL Server 2022 に、実験用テーブルを作成した。この実験用テーブルのデータ件数は 10,000,000 件であり、属性数は 3 である。各属性名は、[id_index],[int_data_index],[float_data] である。各属性の詳細を表 8 に示す。[id_index] は int 型で、1 から 10,000,000 までの値を順番に格納している。[int_data_index] は int 型で、1 から 30 までのランダムな整数値を格納している。[float_data] は float 型で、1 以上 10 未満のランダムな浮動小数点を格納している。[id_index] にクラスター化インデックス、[int_data_index] に非クラスター化インデックスを設定した。このテーブルから、データ件数 N (100 | 200 | 500 | 1,000 | 2,000 | 5,000 | 10,000 | 20,000 | 50,000 | 100,000 | 200,000 | 500,000 | 1,000,000 | 2,000,000 | 5,000,000 | 10,000,000) を取得する。

実験に用いるクエリを図 11 に示す。このクエリでは、WHERE 句によって [id_index] の値を参照してデータ数を N 件に絞り込み、結果データ数を N 件とする。これを、(PC)²DV V2, (PC)²DV, PCPASQL + DV で実行し、クエリ処理時間とデータ描画時間の計測を行った。

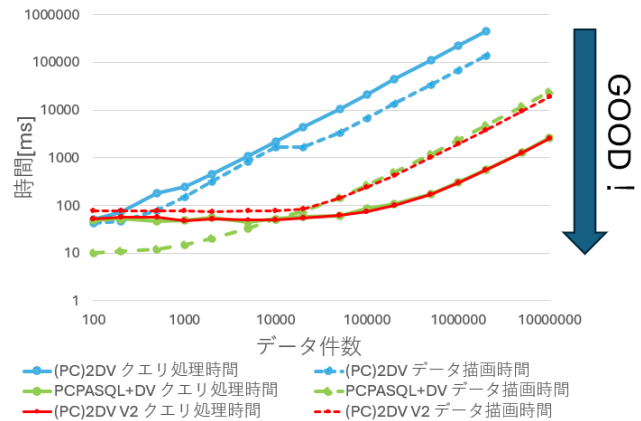


図 12 クエリ処理時間とデータ描画時間の計測結果

6.3 結果と考察

提案システムの (PC)²DV V2 と先行研究である (PC)²DV, PCPASQL + DV における、取得するデータ数に対しての、クエリ処理時間とデータ描画時間を測定した結果を両対数グラフとして図 12 に示す。赤の線が提案システムである (PC)²DV V2 を表しており、青の線が (PC)²DV, 黄緑の線が PCPASQL + DV を表している。また、実線がクエリ処理時間を、点線がデータ描画時間を表している。この図より、(PC)²DV V2 はクエリ処理時間、データ描画時間ともに、速度の改善を行った PCPASQL + DV とほぼ同等となっていることが分かる。これにより、PCPASQL + DV に GUI 操作を追加した (PC)²DV V2 でも、クエリ処理時間、データ描画時間が増えることなく、ビッグデータ分析に有効であることが検証された。なお、今回使用したクエリは図 11 に示すような比較的単純なものであったが、より複雑なクエリを使用する場合には、クエリ処理時間が増加する可能性がある。

7 まとめと今後の課題

我々は、複数の属性からなるデータを PCP により可視化し、SQL を拡張した言語で操作可能な可視化システムを提案してきた。しかし、先行研究である (PC)²DV と PCPASQL + DV にはライフログデータ分析を行う上で、操作後の処理時間が長いことや、GUI による直感的な操作に対応していないといった問題があった。そこで、本研究ではこれらの問題を解決する (PC)²DV V2 を提案した。本システムによる、リレーショナル代数をもとに設計した GUI 操作と、操作結果を保存・再現する情報可視化言語 (PC)²L V2 を用いて、建物の需要電力データの分析を行い、有効性を示した。また、先行研究の実装とのクエリ処理時間とデータ描画時間の定量的な比較を行った。これにより、(PC)²DV V2 によるビッグデータ分析が可能であることを示した。

今後の課題として、散布図や積み上げ棒グラフのような PCP 以外のグラフ描画機能を (PC)²DV V2 に実装し、マルチビューによるデータ分析を可能とすることがあげられる。また、今回の分析で扱った電力データとは全く異なる種類のデータを対象

とした分析により、有用な知見を示していく。

謝 辞

本研究の一部は横浜国立大学人工知能研究拠点学長裁量経費の支援による。

文 献

- [1] 濱崎裕太, 植村智明, 富井尚志. 多変量データを SPJ 質問により統合する平行座標プロット型情報可視化システムと操作言語. 情報処理学会論文誌データベース (TOD), Vol. 12, No. 4, pp. 27–39, October 2019.
- [2] 植村智明, 能條太悟, 吉瀬雄大, 富井尚志. 解析者の興味に基づく道路区間集計が可能な EV 推定消費エネルギーデータ解析システムの構築と応用. 情報処理学会論文誌データベース (TOD), Vol. 14, No. 4, pp. 70–85, October 2021.
- [3] 能條太悟, 杉本航洋, 富井尚志. SQL ライクな操作言語を用いた可視化システムの応用とライフログ分析のためのデータ操作高速化. 日本データベース学会和文論文誌, 第 22-J 巻, pp. 1–10, 2023.
- [4] 木次輝, 宇津木萌, 大久保勇輝, 富井尚志. SQL を拡張した情報可視化言語を用いたライフログデータ分析支援システムのパフォーマンス評価. 第 17 回データ工学と情報マネジメントに関するフォーラム (DEIM2025), 9G-04, pp. 1–8, 2025.
- [5] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, Vol. 1, No. 2, pp. 69–91, 1985.
- [6] Jimmy Johansson and Camilla Forsell. Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, Vol. 22, No. 1, pp. 579–588, 2016.
- [7] G. Grinstein, M. Trutschl, and U. Cvek. High dimensional visualizations. In *In Proceedings of KDD Workshop on Visual Data Mining*, 2001.
- [8] Fatma Bouali, Abdelheq Guettala, and Gilles Venturini. VizAssist: An interactive user assistant for visual data mining. *The Visual Computer: Int'l Journal of Computer Graphics*, Vol. 32, No. 11, pp. 1447–1463, 2016.
- [9] Wenqiang Cui. Visual analytics: A comprehensive overview. *IEEE Access*, Vol. 7, pp. 81555–81573, 2019.
- [10] Takayuki Itoh, Ashnil Kumar, Karsten Klein, and Jinman Kim. High-dimensional data visualization by interactive construction of low-dimensional parallel coordinate plots. *Journal of Visual Languages & Computing*, Vol. 43, pp. 1–13, 2017.
- [11] Z. Zhou, Z. Ye, J. Yu, and W. Chen. Cluster-aware arrangement of the parallel coordinate plots. *Journal of Visual Languages & Computing*, Vol. 46, pp. 43–52, 2018.
- [12] Jinwook Bok, Bohyoung Kim, and Jinwook Seo. Augmenting parallel coordinates plots with color-coded stacked histograms. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 28, No. 7, pp. 2563–2576, 2022.
- [13] Henning Gruendl, Patrick Riehmman, Yves Pausch, and Bernd Fröhlich. Time-series plots integrated in parallel-coordinates displays. *Computer Graphics Forum*, Vol. 35, , 2016.
- [14] Lena Cibulski, Thorsten May, Johanna Schmidt, and Jorn Kohlhammer. Compo*sed: Composite parallel coordinates for co-dependent multi-attribute choices. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, Vol. 29, No. 10, 2023.
- [15] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, Vol. 26, No. 6, pp. 881–906, Dec 2017.
- [16] Manuela Waldner, Stefan Bruckner, and Ivan Viola. Graphical histories of information foraging. *Proc. of the 8th Nordic Conf. on Human-Computer Interaction: Fun, Fast, Foundational(NordiCHI '14)*, pp. 295–304, 2014.
- [17] Peter Mindek, Stefan Bruckner, and M. Eduard Gröller. Contextual snapshots: Enriched visualization with interactive spatial annotations. *Proc. of the 29th Spring Conf. on Computer Graphics(SCCG '13)*, pp. 49–56, 2013.
- [18] Fotis Psallidas and Eugene Wu. Provenance for interactive visualizations. HILDA '18: Workshop on Human-In-the-Loop Data Analytics, 2018.
- [19] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Trans. on Visualization and Computer Graphics(TVCG)*, Vol. 20, No. 12, pp. 2023–2032, Dec 2014.
- [20] Holger Stitz, Samuel Gratzl, Harald Piringer, Thomas Zichner, and Marc Streit. Knowledgepearls: Provenance-based visualization retrieval. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 25, No. 1, pp. 120–130, 2019.
- [21] 増永良文. リレーショナルデータベース入門—データモデル・SQL・管理システム. サイエンス社, 2003.

ファイチューニングの最適化支援のための埋め込み移動軌跡の可視化分析

石川 智也[†] 藤田 秀之^{††} 大森 匡^{††} 新谷 隆彦^{††}

[†] 電気通信大学情報理工学域 〒182-8585 東京都調布市調布ヶ丘一丁目5番地1号

^{††} 電気通信大学情報理工学研究科 〒182-8585 東京都調布市調布ヶ丘一丁目5番地1号

E-mail: [†]ti2210038@edu.cc.uec.ac.jp, ^{††}{fujita,omori,shintani}@is.uec.ac.jp

あらまし 汎用言語モデルでは、テキストやトークンを埋め込みと呼ばれるベクトルで表現する。ファイチューニングの性能評価は、目的とするタスクに対するモデルの精度評価を代表とする、外在的評価が中心だが、精度改善や副作用の根拠を分析することは困難とされている。そこで、埋め込み空間の可視化分析を含む、内在的評価の手法が提案されている。本研究では、ファイチューニング完了後の埋め込み空間だけではなく、ファイチューニングの過程で各埋め込みが移動する軌跡（埋め込み軌跡）に着目し、軌跡マイニングを援用することで、ファイチューニングの過程を可視化分析する手法を検討する。ニュース記事分類タスクに対する BERT ファイチューニングを対象に、各クラスの訓練データ数や F1 スコアの推移と、埋め込み移動軌跡の対応を分析した。特に、学習の進行に従い、埋め込み点の移動速度が低下し、滞留し始めることに着目し、訓練データ数の多いクラスほど移動速度の低下が早く、滞留が多く検出される傾向を確認した。さらに、速度のばらつきを考慮した滞留点検出の手法を提案し、その結果を分析した。分析結果より、埋め込み移動軌跡に対する滞留点検出の活用方針をまとめた。

キーワード 解釈可能性, 汎用言語モデル, 埋め込み可視化, 可視化分析, Human-in-the-loop システム

1 はじめに

ファイチューニングの内在的評価が行われる際、事前学習済みモデルやファイチューニング終了後のモデルにおける埋め込みを可視化し分析することは、これまでよく行われてきた。

それに対し、ファイチューニングの経過における埋め込みベクトルの変化を軌跡として捉え、それを詳細に分析することはあまり行われてこなかった。そのため、埋め込みベクトルの軌跡に対して軌跡データマイニングを適用して分析する手法を検討する。

各クラスの軌跡の類似度などに着目することで、ファイチューニングの有効性をクラスごとやデータごとに判断することができるようになることが期待できる。

2 先行研究

本章では、ファイチューニングの内在的評価を行っている先行研究として、*A Closer Look at How Fine-tuning Changes BERT* [1] について紹介する。言語モデルが、内部でどのように情報を学習しているのか追跡することをプロービングという。この研究では、ファイチューニングの効果を評価するために、モデルを固定特徴抽出器として使いその上にニューラルネットワークを構築することで分類器としての性能を測定する手法と、幾何学的な観点で埋め込みをクラスターリングする DIRECTPROBE [2] を用いて、プロービングを行っている。

DIRECTPROBE は、各クラスが同一ラベルの点のみを含み、クラスターの凸包同士が重ならないクラスターの集合を返す。図 1 に、単純な二値分類問題とそれに DIRECTPROBE を適用

した結果を示す。図 1 から、この 2 クラスの決定境界は、異なるクラスターの間を通るといえることがわかる。DIRECTPROBE を適用した結果、クラスター数とクラス数が等しいとき、その分類問題は線形分離可能な問題だということになる。

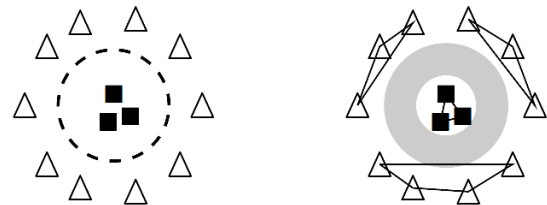


図 1: 単純な二値分類問題とそれに DIRECTPROBE を適用した結果。白い三角形と黒い四角形はクラスを、点同士を繋いだ線はクラスターを表し、左図は破線の円形決定境界を持つ二値分類問題、右図はその問題に DIRECTPROBE を適用した結果を表す。

この研究では、DIRECTPROBE が返すクラスターの集合について、クラスター数、クラスター間距離、空間的類似性を測定し、分析することで、ファイチューニングの内在的評価をしている。

分類器ベースのプロービングと DIRECTPROBE により、ファイチューニング後には DIRECTPROBE のクラスター数が減少し、異なるラベルのクラスター同士の最小距離が増加していくことが確認されている。

また、対象タスクと異なるタスクでファイチューニングを行うクロスタスクファイチューニングでは、対象タスクに類似したタスクでファイチューニングした場合、クラスター数は増加して異なるラベルのクラスター間の距離は増加するが、対象タスクに悪影響を及ぼすと思われるタスクでファイチューニ

ングするとクラス数が増加して異なるラベルのクラス間距離は減少するということがわかっている。

また、上位層であっても埋め込みを無秩序には変更せず、下位層は上位層よりラベルを分離せず、埋め込みの移動範囲が小さいことなどもわかっている。

さらに、この研究では、あるタスクで最も近い3つのラベルの重心の軌跡を追跡している。図2にその3つのラベルの重心の移動軌跡を示す。図2より、ファインチューニングが進むにつれて、上位層では重心は異なる方向に互いに離れるように移動するということがわかる。

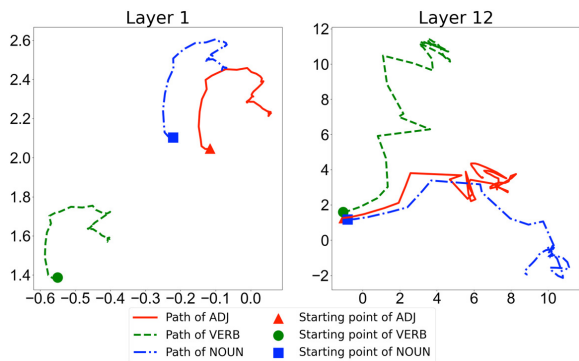


図2: BERT_{base} の POS (Part-of-speech tagging : 品詞タグ付け) で最も近い3つのラベルの重心のファインチューニング中における変化 (PCA 投影)。丸と三角形及び四角形はクラス、線は重心の移動軌跡を表し、左図は第2層の結果、右図は第12層の結果を表す。

3 予備実験

この予備実験では、ファインチューニングの経過に伴って生じる埋め込みの変化を可視化し、それらにどのような傾向が確認できるかを分析する。

ミニバッチを用いてパラメータを更新した回数をステップ数という。本研究では、まず日本語事前学習済み BERT を用いて、テキストをクラスに分類する、分類タスクを対象としたファインチューニングを行う。次に、ファインチューニングを行う際に一定のステップ間隔でモデルを保存していき、保存した各モデルにおける入力テキストの埋め込みベクトルを取得する。そして、取得した埋め込みベクトルから、ファインチューニングの過程における埋め込みベクトルの変化を捉えたアニメーションと軌跡の画像を作成する。

事前学習済みモデルとして、東北大学が Hugging Face 上で提供している事前学習済み BERT を用いた。このモデルは、日本語の Wikipedia を事前に学習している。

データセットには、livedoor ニュースコーパスの Hugging Face Hub 上に公開されたデータセットを使用した。この各データには、記事の「URL、日付、タイトル、本文、カテゴリ」の5つの情報が含まれており、データ件数は6630である。なお、訓練データは5894件、テストデータは736件である。

記事データを分類するタスクを対象タスクとした。このタスクでは、モデルはデータに含まれている「本文」を入力として

「カテゴリ」(以下、クラスと呼ぶ)を出力する。事前学習済み BERT に対して、エポック数を3、モデルを保存するチェックポイントの間隔を10ステップごとにして、上記の分類タスクに向けてファインチューニングを行った。ただし、学習率に関して、スケジューラは線形減衰になっており、Warmup はしない。

埋め込み軌跡に影響を及ぼすと考えられるハイパーパラメータについて説明する。ユーザが設定する学習率に対応してスケジューラが働くとき、エポック数に合わせて学習率が変化する。線形減衰になっている場合は、最大の学習率に達したところから線形に学習率が減少してファインチューニング終了時に0になる。そして、ステップ数を指定して Warmup を利用するときは、学習率が、0からユーザが設定した学習率の最大値まで、指定したステップ数を通じて線形に増加する。また、AdamW は、スケジューラや Warmup の影響を受けた学習率に対し、勾配に適応的な学習率をさらに計算する。この適応的な学習率によってパラメータの変化の大きさは影響を受ける。本研究では、埋め込み点の一定のステップ数あたりの移動距離を移動速度と定義する。パラメータの変化が小さい場合、埋め込み点の移動速度は小さくになると考えられる。

続いて、各チェックポイントで保存したモデルを用いて埋め込みベクトルを取得した。そして、ファインチューニングの経過に伴った埋め込みベクトルの変化を可視化するため、取得した埋め込みベクトルから以下を作成した。

- (i) 各フレームにのテストデータに対する埋め込み空間をプロットしたアニメーション
- (ii) クラス毎に10個データをランダムに選び、その埋め込み軌跡を描画した画像

ただし、(i) と (ii) を作成するにあたり、ファインチューニング終了時のモデルで取得した埋め込みベクトルを PCA で二次元に次元削減して射影行列を求め、その射影行列により他のチェックポイントモデルの埋め込みベクトルも次元削減をした。

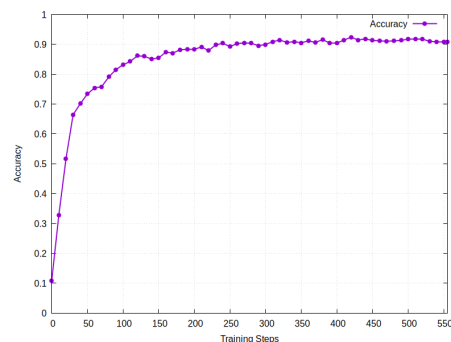


図3: ファインチューニングにおけるステップ数と正解率の関係。縦軸は正解率、横軸はステップ数を表す。

まず、ファインチューニングの過程におけるテストデータに対する正解率の変化を図3に示す。図3から、0ステップ目から100ステップ目までは急激に正解率が高くなっていき、それ

以降は正解率の上昇がやや緩やかになり、300 ステップ目以降はほとんど一定となっていることがわかる。ファインチューニング開始時は正解率は 10.87% であり、440 ステップ目で正解率は最大値の 91.85% となっている。

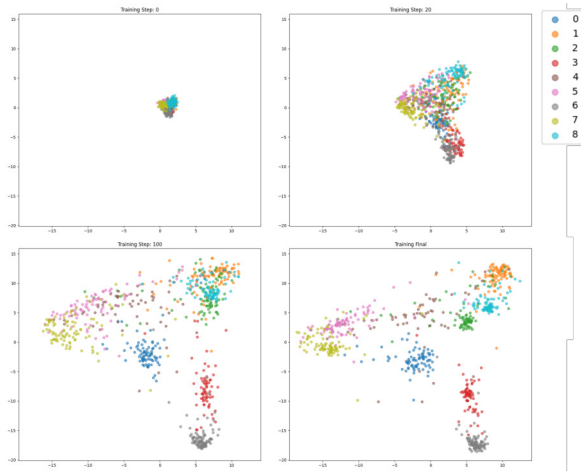


図 4: 埋め込みの移動を表すアニメーションの一部。埋め込み点の色はクラスを表す。左上の図は 0 ステップ目、右上の図は 20 ステップ目、左下の図は 100 ステップ目、右下の図はファインチューニング終了時の埋め込み点の位置を示す。

作成した、(i) のアニメーションの一部を図 4 に示す。図 4 から、図 3 における正解率の変化に対応して、100 ステップ目前後までは埋め込みベクトルの変化は大きいですが、それからは変化が緩やかになっていっているのがわかる。最初は点が原点付近に集まっているが、そこから 100 ステップ目前後までは各点が自クラスの点と共に原点から離れていくように見える。そして、100 ステップ目以降はファインチューニングが進むごとに各点が自クラスの埋め込みの重心に近づいていき、クラスごとに異なる領域に分かれていっているように見える。



図 5: ファインチューニングにおける埋め込み点の軌跡。各色は属するクラスを表し、丸はファインチューニングをする前の点、三角形はファインチューニング終了時の点を表す。

作成した埋め込みベクトルの軌跡を描画した画像を図 5 に示す。図 5 から、やはりファインチューニングの初期段階では

原点付近から離れていく点が多いことがわかる。また、他クラスの点と最後まで一緒に移動する点や、途中まで他クラスの点と共に移動し、やがて自クラスの重心に近づいていく点があることを発見できる。そして、自クラスの重心の近くに存在している点の多くはその場で小さく振動し、逆に自クラスの重心から離れている点の多くはその場で大きく振動しているように見える。

4 提 案

4.1 課題と方針

予備実験により、ファインチューニングが進行すると、多くの埋め込み点が振動し始めるという性質を確認した。埋め込み点の振動を検出するには、軌跡データマイニングにおける滞留点検出が利用できると考えられる。従来の滞留点検出では時間閾値と空間閾値を設定し対象物の速度の低下が一定期間以上続いたと判定されると滞留点として検出する。この従来手法により 5.5 節で埋め込み軌跡における滞留点を検出した。

しかし、埋め込み点はデータによって速度のばらつきがあるため、空間閾値が小さすぎると点が滞留していてもそれを見出せず、空間閾値を大きくしすぎると滞留していない埋め込み点も滞留点として検出してしまう。図 10 の左上の図は空間閾値を $D = 2$ に設定したときの滞留点検出を表しているが、クラス 2 の埋め込み軌跡のような滞留して移動速度が比較的大きい点は滞留点として検出されていない。

埋め込み点は、急激に正解率が向上するファインチューニング初期に原点から離れるように大きい速度で直線的に移動し、やがて速度が低下して振動しているように見える。そこで、埋め込み軌跡のこの特徴を利用し、速度のばらつきを考慮した滞留点検出手法として、直線性指数を滞留点の判定条件に用いた滞留点検出を提案する。

4.2 手 法

まず、提案手法における用語と、滞留点の判定条件について記述する。

埋め込みを取得する頻度を決定する、微小なステップ数のサンプリング間隔を Δt とする。埋め込みを取得する回数を N とし、ファインチューニング開始から Δt ごとの各時刻 t におけるパラメータ集合を θ_t とする。そして、 θ_t に対するデータ d の埋め込みを v_t^d とする。

ウィンドウサイズを k とし、時刻 t から時刻 $t+k$ に対して時間ウィンドウ W を設定して、次式で求めた W における S_t^d が閾値 ϵ より小さいとき、埋め込み v_t^d を滞留点とする。

$$S_t^d = \frac{\|v_{t+k}^d - v_t^d\|}{\sum_{h=t}^{t+k-1} \|v_{h+1}^d - v_h^d\|} \quad (1)$$

滞留点検出の手順について説明する。ファインチューニングにおいて Δt ごとにモデルのパラメータを保存する。クラスの総数を C とし、クラス j ($0 \leq j \leq C-1$) ごとにテストデータセット D から 10 個の記事（データ）をランダムに選択する。選択した各クラス 10 個のデータの集合を D' とする。パラメー

タ集合 $\theta_t (0 \leq t \leq N-1)$ に対する各データ $d (d \in D)$ に対する埋め込み v_t^d を取得する。この際、データ d をモデルに入力した結果の隠れ層の最終層における [CLS] トークンの隠れ状態ベクトルを埋め込みベクトルとして取得する。 S_t^d を求め、データ点 v_t^d が滞留点であるか判定する。

5 実験

埋め込み軌跡に対し、軌跡マイニングの手法を適用し、ファインチューニングを通した埋め込み点の変化を分析する。

本実験では、ステップごとの正解率の推移が異なるクラスが生じるようにするため、不均衡データを用いてファインチューニングを行う。

5.1 ファインチューニングの設定

予備実験と同様に、東北大学の日本語事前学習済み BERT に対し、livedoor ニュースコーパスデータセットを使用して多クラス分類に向けたファインチューニングを行った。

本実験では各クラスの訓練データ数を変更することにより不均衡データにし、テストデータは全クラスでデータ数をすべて均等にした。

訓練データについては、データ数が大 (400 件) のクラス、中 (100 件) のクラス、小 (10 件) のクラスの 3 つのグループを作り、livedoor ニュースコーパスデータセットの 9 クラスをランダムに各グループに 3 クラスずつ割り当てた。その結果、クラス 3, 4, 6 をデータ数大のグループ、クラス 0, 1, 8 をデータ数中のグループ、クラス 2, 5, 7 をデータ数小のグループとした。テストデータについては、アンダーサンプリングを行い、すべてのクラスのデータ数を 45 件とした。

ファインチューニングにおいては、学習率を 2×10^{-5} 、バッチサイズを 32、エポック数を 10 にし、オプティマイザとしては AdamW を使用した。

ただし、Warmup やスケジューラは利用せず、学習率を一定にした。

5.2 F1 スコアの変化

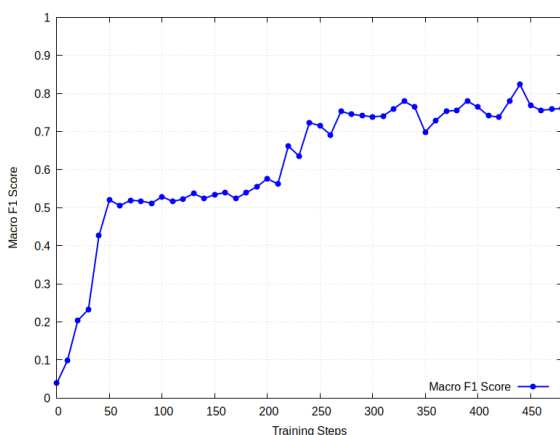


図 6: ファインチューニングにおけるステップ数とマクロ F1 スコアの関係。縦軸はマクロ F1 スコア、横軸はステップ数を表す。

今回は訓練データとして不均衡データをファインチューニングに利用しているため、モデルの性能を評価するために F1 スコアを用いた。まず、ファインチューニングを通したテストデータに対するマクロ F1 スコアの変化を図 6 に示す。図 6 から、マクロ F1 スコアはファインチューニング開始時には 0.0390 であるが、そこから 50 ステップ目の 0.5206 まで急激に増加し、それ以降は 200 ステップ目の 0.5762 まではゆるやかに増加していることがわかる。そして、200 ステップ目から 270 ステップ目の 0.7530 までもう一度激しく上昇し、270 ステップ目からファインチューニング終了時まで 0.6988 から 0.8234 までの範囲で上下していることがわかる。

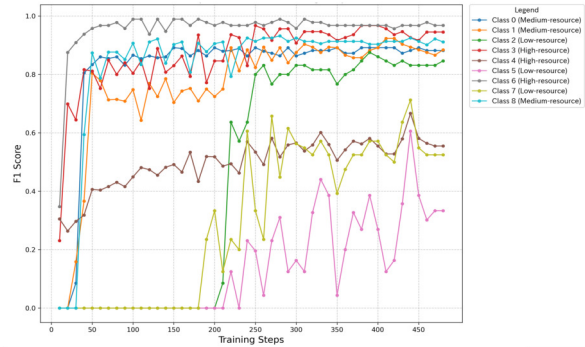


図 7: ファインチューニングにおけるステップ数と各クラスの F1 スコアの変化。各色はクラスを表し、縦軸は F1 スコア、横軸はステップ数を表す。

次に、ファインチューニングの過程における各クラスの F1 スコアの変化を図 7 に示す。図 7 から、データ数の違いに着目すると、データ数の多いクラスほど F1 スコアが最初に急激に上昇するステップ数が小さい傾向があることがわかる。具体的には、データ数大のクラスでは、クラス 3 とクラス 6 は 10 ステップ目から 20 ステップ目にかけて急激に F1 スコアが上昇しており、クラス 4 についても 40 ステップ目から 50 ステップ目にかけて比較的急激に小さいものの上昇している。またデータ数中のクラスでは、概ね 20 ステップ目から 50 ステップ目にかけて急激に F1 スコアが上昇している。そしてデータ数小のクラスでは、概ね 200 ステップ目前後で F1 スコアが急激に上昇している。データ数小のクラスのうち、F1 スコアの最大値が低いクラスは F1 スコアが上下に激しく変動する傾向があることもわかる。具体的には、クラス 4、クラス 5、クラス 7 の F1 スコアは同程度で他クラスと比較すると小さいが、データ数小のクラス 5 とクラス 7 は、データ数大のクラス 4 と比較すると F1 スコアが上下に激しく変動している。図 7 から 9 クラス中、6 クラスは、最終的な F1 スコアが 0.8 を超え、残り 3 クラスは 0.6 を下回っていることがわかる。クラス 4 は、データ数が多いにもかかわらず、F1 スコアが比較的低くなっており、反対にクラス 2 は、データ数が少ないにもかかわらず、F1 スコアが比較的高くなっていることがわかる。以上のことから、データ数によらないクラスごとの本質的な分類難易度が存在することが考えられる。

5.3 軌跡の類似度に基づいたクラスタリング

予備実験と同様に、クラス毎に 10 個データをランダムに選び、その埋め込みベクトルの軌跡を描画した。そして、それら合計 90 個の軌跡データから類似度に基づいてクラスタリングを行い、9 つのクラスに分けた。また比較のため、ファインチューニング終了時の各データの埋め込みの類似度に基づいてクラスタリングを行い、9 クラスに分けた。

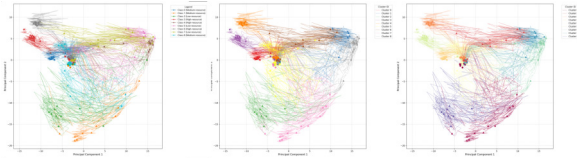


図 8: 埋め込み軌跡のクラスタリングの比較。丸はファインチューニングをする前の点、三角形はファインチューニングを表す。各色は、左の図では属するクラス、中央の図は終了時の埋め込みに基づいて分類されたクラス、右の図では軌跡の類似度に基づいて分類されたクラスを表す。

各埋め込み軌跡の属する正解クラスと、終了時の埋め込みに基づいたクラスタリング結果、軌跡の類似度に基づいたクラスタリング結果を図 8 に示す。予備実験の図 5 では途中まで他クラスのデータ点と共に移動し、やがて自クラスの重心に近づいていくデータ点が見られた。これらはファインチューニングの進行によるモデルの変化を分析する上で重要なデータであると考えられる。予備実験でこのようなデータ点があったことから、本実験において、終了時の埋め込みに基づいたクラスタリングと軌跡の類似度に基づいたクラスタリングで異なるクラスに属するデータ点が存在することを想定していた。しかし、図 8 から、終了時の埋め込みに基づいたクラスタリング結果と軌跡の類似度に基づいたクラスタリング結果が全く同じになっていることがわかる。これは、ファインチューニングがある程度進行すると、埋め込み点が自身の終了時の埋め込み点の近くで滞留し始めることにより、軌跡の類似度が終了時の埋め込み点の影響を強く受けるからであると考えられる。

5.4 埋め込み点の移動速度の変化

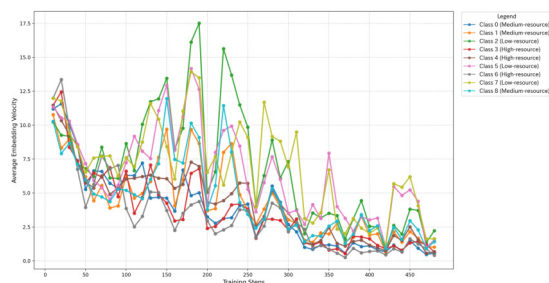


図 9: ファインチューニングにおけるステップ数と各クラスの埋め込み点の移動速度 (10 ステップにおける変位長の平均値) の関係。各色はクラスを表し、縦軸は過去 10 ステップの変位長、横軸はステップ数を表す。

各クラスの 10 ステップごとの埋め込みの移動距離の変化を

図 9 に示す。図 9 から、200 ステップ目以降は全てのクラスで移動速度 (10 ステップあたりの移動距離) が概ね減少していくことがわかる。また、F1 スコアに関係なく、データ数が多いクラスほど移動速度が小さい傾向が見られた。100 ステップ目以降データ数大, 中, 小の順に移動速度が小さい値で概ね推移しており、F1 スコアが低いデータ数大のクラス 4 の移動速度が、F1 スコアが高いデータ小のクラス 2 の移動速度を下回っている。そして、移動速度が低下することにより滞留が始まることを考慮すると、F1 スコアによらずにデータ数が多いクラスほど滞留が早いということがわかる。

5.5 従来手法による滞留点検出

埋め込み軌跡に滞留点検出を適用した。滞留点検出では、空間閾値 D と時間閾値 T を設定し、対象の点 p の移動距離が時間 T の間連続で D 未満のとき、 p を滞留点とする。ここでは、埋め込みを取得するサンプリング間隔ごとの時刻に対し時間閾値を、埋め込みの移動距離に空間閾値を設定する。ただし、サンプリング間隔を 10 ステップごとにして保存しているため、 $T = 4$ のときは 10 ステップ間における埋め込み点の移動距離が 4 回連続で D 未満であるときに滞留とみなされる。

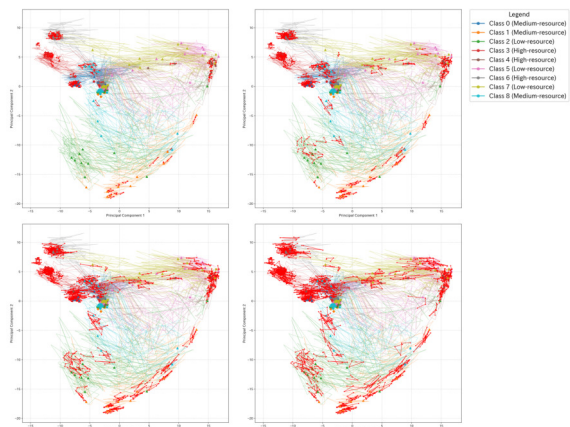


図 10: 移動距離に基づいた滞留点検出。赤色は滞留点として検出された箇所の軌跡、それ以外の各色はクラスを表し、丸はファインチューニングをする前の点、三角形はファインチューニング終了時の点を表す。左上の図が $D = 2$ 、右上の図が $D = 3$ 、左下の図が $D = 4$ 、右下の図が $D = 5$

表 1: 滞留点検出における時間閾値と滞留点検出された軌跡の個数の関係

空間閾値 D	滞留点検出軌跡の個数	滞留点検出軌跡率
2	53	0.599
3	74	0.822
4	87	0.967
5	90	1
10	90	1

$T = 4$ のときの滞留点検出の可視化結果を図 10 に示す。また、空間閾値と、滞留点検出された軌跡 (滞留点検出軌跡)

の個数や全 90 個の軌跡のうちの滞留点検出軌跡の割合（滞留点検出軌跡率）の関係を表 1 に示す。

表 1 より、時間閾値が大きいほど滞留点が検出される軌跡の個数が増加しているのがわかる。特に $D = 5$ になると全ての軌跡で滞留点が検出されている。

また図 7 で F1 スコアの最大値が 0.8 を超えているクラスの中では、図 10 において、データ数小のクラスやデータ数中のクラス 8 を除くと $D = 2$ のときに多くの軌跡で滞留点が検出されていることがわかる。そして、F1 スコアの最大値が比較的小さい 3 クラスの中では、唯一のデータ点大のクラスであるクラス 4 のみ滞留点が検出されていることがわかる。そして、データ数に着目すると、F1 スコアの大小に関係なく、データ数大やデータ数中のクラスでは多くの埋め込み軌跡で滞留点が検出されているのに対し、データ数小のクラスでは滞留点が検出されていないことがわかる。そして、空間閾値が大きくなるほどデータ数小のクラスでも滞留点が検出される軌跡が増加していることがわかる。以上の結果から、F1 スコアの最大値に関係なく、データ数が多いクラスほど、従来手法により滞留点が検出されやすいということがわかる。これは、図 9 においてデータ数が多いほど埋め込み点の移動速度が小さい傾向があり、移動が微小な範囲に留まりやすいからだと考えられる。

ただし、クラス 2 はデータ数小のクラスであり、 $D = 2$ の時点では埋め込み軌跡で滞留が検出されていない。それに対し、 $D = 3$ や $D = 4$ の検出結果で可視化されている検出された滞留点を見るとクラス 2 は多くの軌跡で実際に滞留しているように見える。このことから、データ点ごとの速度のばらつきが大きいために、空間閾値を小さすぎると滞留していても検出できていないのだと考えられる。

5.6 提案手法による滞留点検出

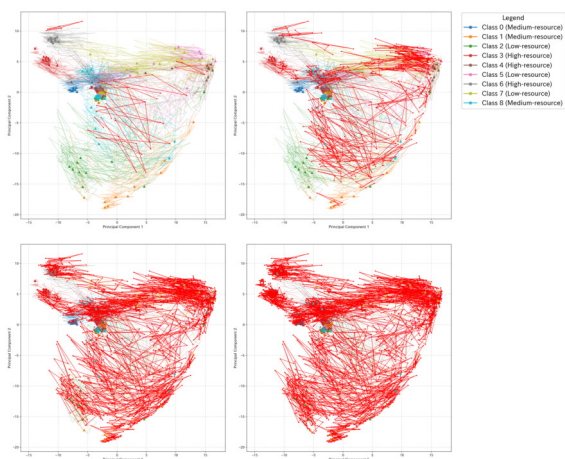


図 11: 提案手法による滞留点検出。赤色は滞留点として検出された箇所の軌跡、それ以外の各色はクラスを表し、丸はファインチューニングをする前の点、三角形はファインチューニング終了時の点を表す。左上の図が $\epsilon = 0.1$ 、右上の図が $\epsilon = 0.15$ 、左下の図が $\epsilon = 0.2$ 、右下の図が $\epsilon = 0.25$ のとき。

ウィンドウサイズ k を 5 とし、提案手法による滞留点検出を

行った。チェックポイントを 10 ステップごとにしてモデルを保存しているため、埋め込み点を滞留点か判定する際には、50 ステップ先の点との直線距離を 50 ステップ先までの総距離で割ったものを直線性指数として求める。提案手法による滞留点検出の可視化結果を図 11 に示す。また、直線性指数に対する ϵ と、滞留点検出軌跡個数や滞留点検出軌跡率の関係を表 2 に示す。

表 2: 滞留点検出における時間閾値と滞留点が検出された軌跡の個数の関係

閾値 ϵ	滞留点検出軌跡の個数	滞留点検出軌跡率
0.1	27	0.3
0.15	41	0.456
0.2	71	0.789
0.25	90	1

表 2 より、 ϵ が大きいほど滞留点が検出される軌跡の個数が増加していて、 $\epsilon = 0.2$ になると全ての軌跡で滞留点が検出されている。また図 11 から、 $\epsilon = 0.1$ では、滞留として検出されている軌跡の中にほとんど滞留とみなせるものがないということがわかる。また $\epsilon = 0.15$ や $\epsilon = 0.2$ 、 $\epsilon = 0.25$ になると滞留とみなせるものも検出されているが、そうでないものも同時に多く検出されていることがわかる。以上のことから、提案手法による滞留点検出は誤検出が非常に多いと考えられる。

5.7 直線性指数の変化

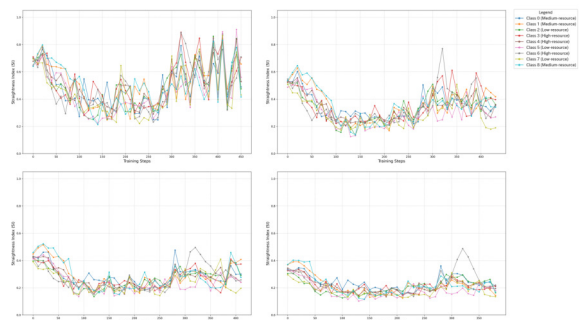


図 12: ファインチューニングにおけるステップ数と各クラスの直線性指数の平均値の変化。各色はクラスを表し、縦軸は直線性指数、横軸はステップ数を表す。左上の図が $k = 3$ 、右上の図が $k = 5$ 、左下の図が $k = 7$ 、右下の図が $k = 10$ のとき。

ファインチューニングにおける直線性指数の変化を図 12 に示す。図 12 から、すべてのウィンドウサイズについて、全てのクラスの直線性指数が、0 ステップ目から 100 ステップ目にかけて概ね減少していき、そこから 250 ステップ目までは大局的には値があまり変化せず、250 ステップ目から 300 ステップ目にかけて増加していき、その後また値が大局的に保たれるという変化の仕方をしてしていることがわかる。また、ウィンドウサイズが大きいくほど直線性指数のとりうる範囲が小さく、変化もゆるやかであることがわかる。そして、300 ステップ目以降は学習が進行し、F1 スコアが安定しているクラスが多いが、直線性指数は下がらず、比較的高い値となっていることがわかる。以上

のことから、F1 スコアの推移が異なるクラスが存在するデータセットであるにもかかわらず、全クラスで概ね共通した性質を持っていることがわかる。このことから、直線性指数を判定条件に用いた提案手法により誤検出をせず滞留点を検出することは困難であると考えられる。

6 まとめと今後の活動方針

本研究では、ファインチューニングにおける埋め込み点の移動軌跡に着目し、ファインチューニング過程を分析した。

それにより、多くの埋め込み軌跡にはファインチューニングが進行すると滞留し始めるという性質があることを発見した。そこで、埋め込み軌跡の類似度は、滞留によりファインチューニング終了時の埋め込み点の影響を強く受けてしまうという課題も確認された。埋め込み軌跡から滞留している部分を取り除いたり、埋め込み軌跡の簡略化したりすることで対応できると考えられる。

そして、埋め込み軌跡の特徴から、直線性指数を利用して、埋め込み軌跡に特化した滞留点検出手法を提案した。この提案手法で実際に滞留点検出をすると、誤検出が多い結果となった。今後、速度のばらつきを考慮した滞留点検出をできるように、手法を改善していく必要がある。

さらに、埋め込み軌跡に対する分析結果から、データ数が多いクラスほど F1 スコアの高さに関係なく埋め込みの移動速度が小さくなりやすいという性質を確認した。また、データ数が少ないクラスは滞留点検出では滞留点の検出が困難であるということも確認した。このことを踏まえて、ファインチューニングの最適化支援に向けた滞留点検出の活用方針をまとめる。滞留点検出を利用することで、分類性能スコアのみでは識別できない、データを追加しても性能向上しないクラスを識別可能であると考えられる。滞留が早いクラスの場合、分類性能が高ければ、そのクラスは学習する上で理想的な状態にあるといえるが、分類性能が低ければ、学習が本質的に難しい状態にあると想定できる。このような滞留が早く分類性能が低いクラスは特徴量の再設計やラベルの再定義により改善できる可能性がある。そして、滞留が遅いクラスの場合、分類性能が高ければ汎化性能が低く分類性能が低ければデータ不足であると考えられる。このような滞留が遅いクラスは重点的なデータ追加による改善できると考えられる。今後、この活用方針に従って、ファインチューニングの最適化支援を行う実験をする予定である。

文 献

- [1] Yichu Zhou and Vivek Srikumar. A closer look at how fine-tuning changes bert. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1046–1061, 2022.
- [2] Yichu Zhou and Vivek Srikumar. DirectProbe: Studying representations without classifiers. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5070–5083, 2021.

不動産情報探索VRインタフェースにおける属性分布可視化

中山 裕紀[†] 大島 裕明[†]

[†] 兵庫県立大学情報科学研究科 〒 651-2197 兵庫県神戸市西区学園西町

E-mail: [†] yuki.nkym256@gmail.com, ohshima@ai.u-hyogo.ac.jp

あらまし 本研究では、不動産情報探索における俯瞰的閲覧のための、属性間の条件分布を可視化して提示するインタフェースを提案する。不動産情報は、賃料や間取りなど多くの属性を持ち、各属性が取り得る条件が存在する。不動産情報探索は、多くのユーザが明確なニーズを持たず、検索を通して得た情報をもとに求める条件を明確にする必要がある。そのためには、最寄り駅ごとの賃料分布など、属性間の条件分布を提示することが有効である。そこで本研究では、属性間の条件分布を可視化し、検索結果と同時に提示する手法を提案する。提案するインタフェースでは、不動産情報は1つの属性について、条件ごとに集約して表示する。各条件について、他の属性の条件分布を積み上げ棒グラフによって可視化する。また、インタフェース上で表示する不動産物件について、条件ごとに色分けを行い提示する。本論文では、この手法によってユーザが属性間の条件ごとの分布をどの程度理解することができているか評価した。

キーワード VR, インタフェース, 探索的検索

1 はじめに

近年、不動産物件を探す際、不動産情報検索サイトが多く用いられる。これら不動産情報検索サイトの多くは、検索結果を並べてリスト状に表示している。不動産情報の検索結果は、数百件を超える多量なものになる場合もある。このような多量の検索結果をリスト状に表示した場合、全ての検索結果を閲覧することが難しいという問題がある。また、不動産情報の検索は、多くの場合ユーザに明確な検索条件が存在せず、ユーザは提示された検索結果を閲覧しながら検索条件を明確にしていく。例えば、最寄り駅ごとの賃料分布や、賃料と間取りのトレードオフの関係性などの属性間の条件分布を検索結果から把握し、ニーズを満たす検索条件を明確にする。この際、リスト状の表示では多数の検索結果を閲覧し、結果がニーズと合わない場合はクエリを更新してさらに閲覧する必要がある。このように、多量な不動産情報をリスト状に表示することには複数の問題が存在する。その解決には、多量の不動産情報を一度に閲覧でき、かつ属性間の条件分布を一目で把握できることが有用である。そこで、本研究では不動産情報の全体像を俯瞰的に閲覧可能とする Virtual Reality (VR) インタフェースを提案する。

VRを用いると、仮想空間に没入し、仮想空間上に配置された情報がまるでその場にあるかのように閲覧することが可能になる。仮想空間は広大な領域を持ち、自由に情報を配置することができる。本研究では仮想空間の広大な領域に着目し、多量の不動産情報の表示に活用する。

本研究では、不動産情報の賃料や間取りなどを「属性」と呼ぶ。また、賃料における10,000円以上20,000円未満、間取りにおける1K、1DKなど、各属性が取りうる値やその範囲を「条件」と呼ぶ。「不動産物件」とは、不動産情報に含まれる個々のデータを指す。

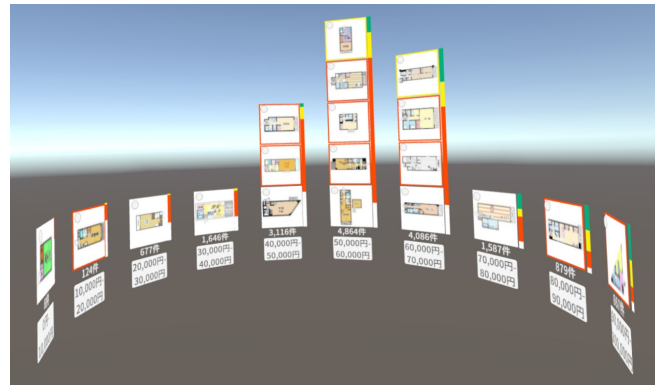


図1 不動産情報の表示

本研究で提案するシステムでは、不動産情報を条件によって複数のグループに分け、グループごとに集約して表示する。提案システムで、賃料の条件をもとにグループ分けを行い不動産情報を表示した様子を図1に示す。各グループの不動産物件数をもとに、配置するパネルの枚数が変化する。不動産物件数が多いほど、パネルの枚数は多くなる。このように表示された不動産情報を、VRを用いて俯瞰的に閲覧することで、不動産情報の全体像を把握することが可能である。また、提案システムは不動産情報の絞り込み、提示情報の詳細度 (Level of Detail: LoD) 制御、不動産物件のお気に入り登録、属性間の条件分布可視化という4つの機能を持つ。これらの機能により、ユーザは不動産情報の全体像から、ニーズに合う条件を設定して絞り込みを行うことが可能である。

本論文では、提案するインタフェースによってユーザが属性間の条件分布を理解することができるかを明らかにする。そのために、提案システムと平面システムをユーザ実験により比較する。各システムを用いて不動産情報探索を実施した後、属性間の関係性について問うクイズとアンケートを実施し、被験者

の理解度を評価する。

2 関連研究

本節では、以下に示す3つの項目について先行研究を引用し、本研究の位置づけを示す。

- VRを用いた検索
- 探索的検索とインタフェース
- 不動産情報検索

2.1 VRを用いた検索

検索インタフェースにVRを用いる研究は広く行われている。Wardら[1,2,3]はVRを用いた検索における効果的な検索結果の提示方法について、複数の観点から検証している。まず紹介する論文[1]では、検索結果を1列に8件表示するリスト表示、4行5列でのグリッド表示、2行8列の円弧状表示の3通りの手法について、ユーザ実験により比較している。ユーザ実験の結果、目標となる検索結果の発見までにかかる時間は円弧状表示が最も速い一方、主観的な評価では、リスト表示やグリッド表示が好まれたことを示している。本研究で提案するインタフェースはこの円弧状表示を採用している。

また、別の論文[2]では、VRを用いた商品検索における、検索環境の文脈整合性と検索結果の提示方法の違いが、ユーザの体験に与える影響を調査している。同論文では、ユーザが実際に検索を行っている物理的環境と、VR内に表示される検索環境との差を検索環境の文脈整合性と定義している。また、提示する検索結果の詳細度について3通りの手法を比較している。これにより、検索環境と現実空間の類似度と、検索結果提示の詳細度が、検索時の行動や検索の満足度に与える影響を調査している。実験により、現実の文脈を再現した検索環境と、詳細な検索結果提示の組み合わせにおいて、ユーザの検索結果への満足度や検索の効率が最も高くなることを示している。

さらに別の論文[3]では、VRを用いた4種類の検索結果提示手法について、ユーザの探索行動や好みを調査している。同論文では、縦に並べるリスト表示、 3×3 や 4×4 のグリッド表示を比較し、結果としてユーザが「リスト表示は労力を要する」と認識する傾向にあることや、グリッド表示の方が順序にとらわれず、より下位に表示された検索結果まで選択されたことを明らかにしている。

Takeuchiら[4]は、VR環境での検索結果の提示において、検索結果を空間的に配置する「Spatial Bookmark」システムを提案し、ユーザの検索結果の記憶へ与える影響を検証している。同論文では、ユーザに検索タブや検索結果を、4種類の空間に配置させ、テストによって検索した内容をどの程度記憶しているか調査している。その結果、家具などを配置した現実的な空間で、ユーザの記憶が最も定着することを示している。

Giunchiら[5]はVR環境における3Dモデルの検索手法として、ベースとなる3Dモデル上にスケッチを描きこむことで、形状や色などの特徴をクエリとして入力する手法を提案している。同論文ではユーザ実験により、既存のリストをスクロール

する手法と提案手法を比較し、検索完了までの時間やユーザの好みの観点で提案手法の方が優れていることを示している。

SchleuBinger [6]は情報検索のためのVRインタフェースを提案した1,042件の論文について調査、分析を行っている。その結果、多くの論文で従来のリスト状表示ではない手法が提案されており、その種類は多岐にわたることを示している。また、評価方法については従来手法との比較を行っている論文が多く、そのほとんどで従来の手法より検索実行時間が短く、検索結果の正答率が高くなることを示している。

VRなどの没入型環境においてデータの理解や意思決定を支援する研究が行われている。Dwyerら[7]は「Immersive Analytics」という研究分野を提唱している。Immersive Analyticsは2Dでは表示、閲覧が難しい3D散布図などをVRやARといった没入型環境内で閲覧し、ユーザの理解を支援することを目的とする。

Coffeyら[8]は3次元のボリュームデータを探索するためのVRインタフェースである「Slice WIM」を提案している。同手法は「Overview+Detail」を取り入れ、手元のモデルで全体像を俯瞰しつつ、大画面で詳細を確認することを実現している。

Yangら[9]は没入空間での3D散布図の探索において、概観表示とズーム機能の有無について4通りの探索手法を比較し、適した手法を調査している。結果としてVRを用いた閲覧において概観表示機能とズーム機能は、両方ない場合と比べて応答時間、正解率ともに向上させるという結果を示している。概観と詳細を同時に提示する手法のVRにおける優位性が先行研究で示されており、本研究でもこの手法を取り入れる。

2.2 探索的検索とインタフェース

不動産情報の検索において、ユーザは明確な目標を持って検索するわけではなく、検索結果の閲覧やクエリの更新を通じて目標となる不動産物件を定めていく。このような検索行動は探索的検索[10]と呼ばれる。探索的検索における効果的なインタフェースについての研究が広く行われている。

Hoeber[11]は、探索的検索におけるインタフェースの5つの設計原則と、それに基づくインタフェースの評価指標を提案している。この設計原則とは、軽微な追加、視覚的な支援、解読可能性、対話性、永続性である。同論文では、25本の論文に対し、設計原則に基づいた分類、分析を行い、その有効性を検証している。分析の結果、多くの論文で軽微な追加や視覚的なアプローチを採用している一方、永続性に関して取り組んだ研究が少ないという傾向や課題を明らかにし、有効性を示している。本研究では同論文における視覚的なアプローチを採用したインタフェースを提案する。

Mehdiら[12]は、探索的検索における情報の可視化が、絞り込みや情報の理解をどのように支援するのか、文献レビューにより調査している。同論文では探索的検索を支援する手法として、スライダー操作によって即座に検索結果を更新する動的クエリや、検索結果の分布を棒グラフで可視化する手法などがあると分析している。

Medlarら[13]は探索的検索におけるクエリ提案の手法とし

て「代替クエリ」を提案している。代替クエリは現在の検索結果表示と類似する表示が行えるようなクエリを動的に生成し、各クエリの重要性をヒストグラム状に提示する手法である。ユーザ実験ではユーザが提示されたクエリを積極的に用いることや、この手法が新たなクエリの提案だけでなく、検索結果の要約として機能していることを示している。

Crescenzi ら [14] は探索的検索における情報の整理、統合やメタ認知を支援する「OrgBox」の効果を検証している。OrgBox はドメインに依存せず、検索中に発見した情報を整理、構造化することができるツールである。一般的なテキストエディタを模したツールと比較し、ユーザのタスク難易度への認識は変化しないまま、タスクを支援することを示している。

吉田ら [15] は、検索ブラウザの探索履歴をグラフ構造として可視化するシステムを提案し、それがユーザの探索的検索における知識構造の充実化に有用であることを示している。同論文では、閲覧した Web 記事をノード、クエリなどの遷移関係をエッジとするグラフ構造により、Web 検索履歴を可視化し、検索履歴の振り返りを支援するシステムを提案している。ユーザ実験の結果、探索中に閲覧した情報間の関係性をユーザに提示することで、ユーザが構造的に知識を獲得できることを示している。

データ探索を目的とした可視化手法について研究が行われている。Wongsuphasawat ら [16] は情報可視化におけるチャート等の作成を、自動的に推薦する「Voyager」というシステムを提案している。このシステムはユーザの注目している変数について、他の変数との組み合わせや可視化手法を自動的に推薦し、それらをギャラリー状に表示する。同論文では手動可視化ツールと比較し 2 倍以上のデータの組み合わせをユーザが比較しており、データの網羅的な探索に貢献していることを示している。本研究では不動産情報について多様な属性を組み合わせで分布を可視化し、不動産情報の網羅的な閲覧を支援する。

2.3 不動産情報検索

不動産情報検索におけるユーザへの情報提示手法について多くの研究が行われている。Li ら [17] は多様な不動産情報の知識レベルのユーザに対し、物件探しと不動産市場データ分析の両方を支援する「HomeSeeker」を提案している。提案システムは不動産物件の価格データと周辺施設の情報を統合し、知識を持たないユーザが市場を学習しながら不動産物件のニーズを学習することを可能にする。同論文では、ケーススタディにより提案システムを用いることで、既存システムでは難しい価格の地域差の発見が可能であることを示している。

Sun ら [18] は、高次元で複雑な時間、空間パターンを持つ不動産市場のデータを分析するため、複数の可視化技術を統合した視覚的分析システムを提案している。提案システムは地図、積み上げグラフ、ピクセルバー、ツリーマップという 4 つの「Coordinated Views」を連携させ、ユーザは柔軟に市場の動向や物件の属性を探索できる。

諏訪ら [19] は不動産情報の探索における「静けさ」や「日当たり」などの定性的で比較が困難な条件について、定量的に比

表 1 抽出した不動産情報データの属性

物件 ID	賃料	管理費	礼金	敷金
間取り	最寄り駅	最寄り駅距離	築年数	面積
住所	建物構造	物件の特徴		

表 2 抽出した不動産物件と画像データの数

データの種類	能勢電鉄沿線	神戸電鉄沿線
抽出した不動産物件数	6,308	6,136
抽出した画像データ数	114,820	101,775

較可能な指標を構築し、その性能を評価している。同論文では静音性、防音性、採光性、断熱性という 4 つの評価指標についてセンサを用いて定量化するシステムを構築している。空き家に対して実施した評価により、防音性を除いた 3 つの指標について有効であることを示している。

不動産情報の属性は、部屋数が多くなるほど家賃が高くなる、などといったトレードオフの関係性を持つ場合がある。Milutinovic [20] らは、地理空間上でトレードオフの関係性がある意思決定を支援する手法を提案している。同論文では、属性の散布図と地図を連動させることで、属性値どうしの関係性だけでなく、地図上の位置に与える影響も可視化する手法を提案している。

湯本 [21] は「賃料が高くてもいいから駅に近い物件に住みたい」といったユーザのトレードオフな要求に対応する不動産物件の推薦システムを提案している。同論文では条件検索による絞り込みとラフ集合によるルール抽出によって、ユーザの潜在的な好みを反映した推薦を可能にしている。本論文では不動産情報の推薦ではなく、トレードオフな関係性の可視化というアプローチによって同様の課題に取り組む。

3 データセット

3.1 LIFULL HOME'S データセット

提案システムの実装にあたり、「LIFULL HOME'S データセット¹」を利用した。LIFULL HOME'S データセットは、国立情報学研究所の IDR データセット提供サービスにより、株式会社 LIFULL から提供を受けたデータセットである。この中から LIFULL HOME'S 賃貸物件スナップショットデータを使用した。このデータセットには総数 5,334,370 件の不動産物件について、計 71 個の属性と各不動産物件の画像データが含まれている。画像データは 1 件の不動産物件あたり複数枚含まれることがあり、20 枚を超える場合もある。画像データは「間取り図」、「外観写真」、「地図」、「周辺画像」、「内装」、「風呂」、「玄関」、「居間」、「キッチン」、「寝室」、「子供部屋」、「その他」の 12 種類の画像種別に分けられる。データセット内の画像データの総数は 83,403,687 件である。

1: 株式会社 LIFULL (2015): LIFULL HOME'S 賃貸物件スナップショットデータ. 国立情報学研究所情報学研究データリポジトリ. (データセット). <https://doi.org/10.32130/idr.6.1/>



図 2 前処理後の画像



図 5 お気に入り物件のパネル

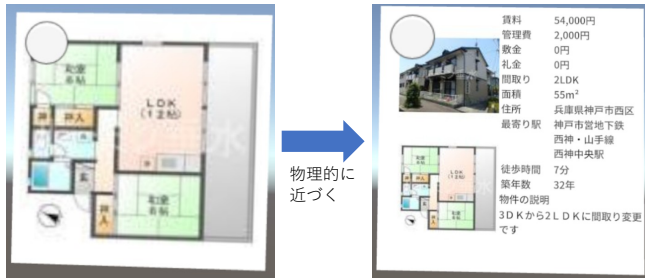


図 3 左：離れた状態の表示、右：近づいた状態の表示



図 4 お気に入り物件の表示

3.2 データセットの前処理

データセットを利用するにあたり、不動産情報データおよび画像データに対し前処理を施した。不動産情報データには、物件種別がマンションやアパートなど居住用賃貸のものと、店舗や事務所など事業用賃貸のものが存在する。この中から物件種別が居住用賃貸の不動産情報を抽出した。次に、最寄り駅が能勢電鉄沿線、および神戸電鉄有馬線沿線に神戸高速鉄道の新開地駅を含めた駅である不動産情報を抽出した。71個の属性のうち、表1に示す属性を抽出した。また、抽出した各不動産物件の画像データを抽出した。なお、画像データが一枚も含まれない不動産物件は取り除いた。このようにして抽出された不動産物件数と画像データ数を、表2に示す。画像データのサイズは、長辺が最大120ピクセルで、短辺のサイズは定められていない。そこで、辺のサイズが120ピクセルの白色正方形を作成し、その中心に画像データを配置することで、120×120ピクセルの画像に変換した。前処理後の画像データを図2に示す。

4 不動産情報探索 VR インタフェース

4.1 提案システムの概要

本節では提案システムが持つ機能の概要を説明する。提案シ

ステムでは、不動産情報を図1に示すように表示する。不動産情報は条件ごとに複数のグループに分けられ、グループごとに集約して表示される。不動産物件は正方形の「不動産情報パネル」上に表示される。不動産情報パネルには複数件の不動産物件が集約され、一定時間ごとに切り替わり表示される。グループに含まれる不動産物件数の割合に基づき、そのグループの不動産情報パネルの枚数が変化する。

提案システムは、この基本となる俯瞰的表示機能に加え、以下の4つの機能を持つ。各機能のより詳細な説明は、著者らの先行研究 [22] を参照されたい。1つ目は表示する不動産情報の条件を絞り込む「絞り込み」である。ユーザは表示している不動産情報の中からグループを選択し、選択したグループの条件を満たす不動産情報のみを表示することができる。属性を変更した場合においても絞り込んだ条件は保持されるため、多様な属性について条件を絞り込み、ユーザのニーズを満たす不動産情報を表示することが可能である。また、絞り込まれた条件に従い、グループをより細分化して表示する。

2つ目は不動産物件の情報詳細度を、ユーザと不動産情報パネルとの物理的な距離により変更する「LoD制御」である。ユーザが表示情報から物理的に離れた位置で俯瞰的に閲覧している場合、間取り図のみの詳細度が低い情報を表示する。ユーザが表示情報に物理的に近づいている場合、賃料や間取りなどを加えた詳細度が高い情報を表示する。それぞれの状態における表示を図3に示す。詳細度が高い状態では、不動産物件の詳細な情報を長時間閲覧できるように、不動産物件の切り替えを停止する。この機能により、俯瞰的に閲覧している時と不動産物件の詳細な情報を閲覧している時で、適した詳細度へと動的に切り替えることが可能である。

3つ目は不動産物件を登録し、それらのみを表示するように切り替える「お気に入り」である。提案システムは不動産物件をお気に入り物件として登録する機能を持ち、不動産情報探索の過程で住みたいと感じた不動産物件を保持することができる。登録したお気に入り物件は、俯瞰的表示から切り替えて、図4のように表示する。お気に入り物件は1枚のパネルに1件表示し、一定時間ごとに切り替わることはない。お気に入り物件を表示するパネルを図5に示す。お気に入り物件は、ユーザが指定したお気に入り物件どうしの配置箇所を入れ替える機能により、自由に並び替えることが可能である。

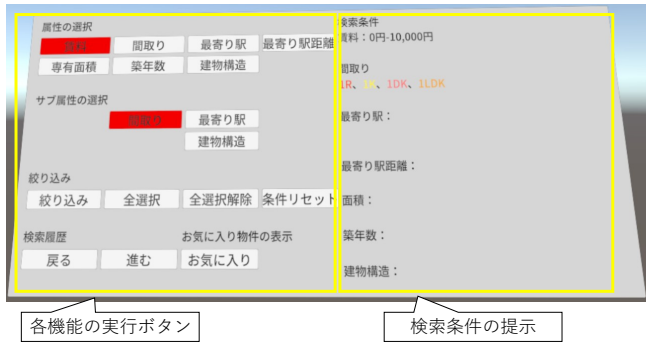


図 6 各機能を実行するメニュー

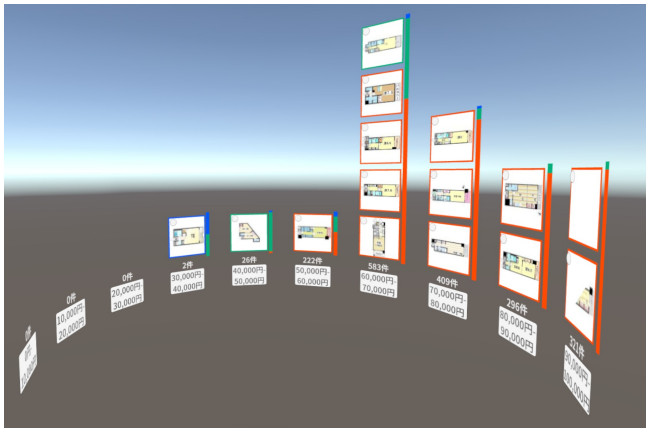


図 7 色分けと積み上げ棒グラフによる複数属性間の条件分布可視化

4つ目は賃料の条件ごとの最寄り駅の傾向といった複数の属性を、不動産情報パネルの色分けや積み上げ棒グラフにより提示する「属性間の条件分布可視化」である。本機能については以下の節で詳細を説明する。

各機能の実行には図 6 に示すようなメニューを用いる。このメニューは左側に配置された各機能の実行ボタンと、右側に表示する検索条件の提示部で構成される。

4.2 属性間の条件分布の可視化

提案システムは賃料条件ごとの各最寄り駅の割合など、各条件における別の属性の条件分布を可視化する機能を持つ。本研究では、この手法として不動産情報パネルの色分けと積み上げ棒グラフの提示を提案する。色分けと積み上げ棒グラフを提示した様子を図 7 に示す。

4.2.1 可視化属性の選択と条件への色割り当て

ユーザは不動産情報をグループ分けする属性に加えて、可視化のための属性を選択可能である。ここで選択できる属性は間取り、最寄り駅、建物構造といった質的な属性である。

可視化のための属性における各条件に対し、ユーザは色を割り当てることができる。色の割り当ては、各グループの下部に表示されるカラーパレットを用いて行う。カラーパレットで色を選択すると、そのグループの条件に色が割り当てられる。色の割り当ての様子を図 8 に示す。複数の条件をまとめたグループの場合、両方の条件に選択した色が割り当てられる。例えば、図 8 の場合では、総合運動公園駅と学園都市駅の両方の条件に



図 8 条件ごとの色の割り当て

赤色が割り当てられる。

4.2.2 色分けと積み上げ棒グラフの提示

属性分布の可視化手法の一つとして、不動産情報パネルの緑に割り当てられた色を適用する色分け機能を提案する。不動産情報パネルに表示されている不動産物件の条件が、色を割り当てられた条件と同じ場合に、不動産情報パネルの縁にその色を適用する。色が割り当てられていない条件の場合、緑の色は変わらず白色のまま表示する。この機能により、ユーザは表示される不動産物件の条件を一目で把握することが可能である。

また、積み上げ棒グラフを用いて、可視化のための属性における各条件の不動産物件数の割合を提示する手法を提案する。積み上げ棒グラフは各グループの右隣に表示される。積み上げ棒グラフ全体の高さは、各グループの不動産情報パネルを積み上げた高さと同様になる。可視化のための属性における各条件に割り当てられた色は、グループにおける各条件の物件数割合と同様になるよう表示する。色が割り当てられていない条件の場合、積み上げ棒グラフ中では白色で表示する。

5 評価実験

5.1 実験の概要

本研究では、提案する VR システムの俯瞰的閲覧により、ユーザが属性間の条件分布を理解できているか調査するため評価実験を行う。評価実験には大学生 4 名が参加し、全員 VR の使用経験はほとんどないと回答した。

評価実験では、被験者には VR システムと、比較のための平面システムの両方を用いて不動産情報探索タスクに取り組みせ、その後クイズとアンケートに回答させる。不動産情報探索タスクでは、被験者には物件を探す条件を与え、その条件を満たす不動産物件を 1 件選択させる。提案システムと平面システムで、探索する不動産情報は異なる路線沿線のデータを用いる。両システムでの不動産情報探索タスク完了後、クイズとアンケートを実施し、被験者の属性間の条件分布理解度を測る。

5.2 比較用平面システム

VR システムとの比較手法として、既存の Web インタフェースを模した平面システムを実装した。平面システムの表示例を図 9 に示す。平面システムでは、検索結果はリスト状に表示され、VR システムと同様に絞り込み、検索結果のソート、お気に入りの機能を持つ。



図 9 平面システムの画面

表 3 被験者グループの分け方

	先に提案システム、 後に平面システム	先に平面システム、 後に提案システム
先に能勢電鉄、 後に神戸電鉄	先に提案システムで能勢電鉄、 後に平面システムで神戸電鉄	先に平面システムで能勢電鉄、 後に提案システムで神戸電鉄
先に神戸電鉄、 後に能勢電鉄	先に提案システムで神戸電鉄、 後に平面システムで能勢電鉄	先に平面システムで神戸電鉄、 後に提案システムで能勢電鉄

表 4 各路線における設問の内容

路線	設問番号	設問
能勢電鉄	Q1	指定した条件では大学最寄り駅（一の鳥居駅）周辺にどれくらい多くの物件が存在するか
	Q2	指定した条件における間取りごとの物件数の傾向について述べた選択肢の文章から正しいものを選べ
	Q3	最寄り駅が川西能勢口駅で面積の広い家を探すことを想定し、最寄り駅と面積の関係について述べた選択肢の文章から正しいものを選べ
	Q4	指定した条件では、40,000 円台に 1LDK の物件はどれくらい多く存在するか
神戸電鉄	Q5	指定した条件において選択肢の駅で最も物件数が多いのはどの駅か
	Q6	指定した条件における賃料ごとの物件数について述べた選択肢の文章から正しいものを選べ
	Q7	指定した条件において、築年数が 10 年以上 15 年未満の築浅物件の最寄り駅の傾向について述べた選択肢の文章から正しいものを選べ
	Q8	築年数が 10 年以上 15 年未満の築浅物件で、間取りが 1DK か 1LDK の広い物件を探す際、このような条件の物件の見つかりやすさについて、情報探索を通じた感覚をもとに選択肢の文章から正しいものを選べ

5.3 不動産情報探索タスク

被験者に取り組みさせる不動産情報探索タスクについて説明する。被験者は VR システムと平面システムを両方用いて本タスクに取り組む。システムごとに最寄り駅の路線が能勢電鉄、または神戸電鉄有馬線のいずれかの不動産情報を探索する。用いるシステムや探索する不動産情報の順序効果を考慮し、被験者を 4 つのグループに分ける。グループの詳細を表 3 に示す。

表 5 SUS の質問（1：そう思わない，5：そう思う）

質問項目
このシステムを頻繁に使用したいと思う
このシステムは不必要に複雑だと感じた
このシステムは使いやすいと思う
このシステムを使うには技術的な支援が必要だと思う
このシステムの中の各機能はうまく統合されていると思う
このシステムには不整合が多いと感じた
ほとんどの人はこのシステムの使い方をすぐに覚えられると思う
このシステムは使いにくいと感じた
このシステムを自信をもって使いこなせる
このシステムを使い始める前に多くのことを学ぶ必要があった

本タスクでは、被験者には以下の条件のもとに住む不動産物件を 1 件選択させる。

- ・ 状況：大学生の一人暮らし
- ・ 予算：80,000 円以下の物件
- ・ 間取り：ワンルーム，1K，1DK，1LDK のいずれか
- ・ 能勢電鉄の最寄り駅：川西能勢口駅，絹延橋駅，平野駅，一の鳥居駅，畦野駅
- ・ 神戸電鉄の最寄り駅：新開地駅，湊川駅，鶴越駅，鈴蘭台駅，北鈴蘭台駅
- ・ 最寄り駅からの距離：徒歩 15 分以内

被験者には各路線について、都心部との接続が良い駅と大学最寄り駅を状況として与えた。能勢電鉄の場合、都心部との接続が良い駅は川西能勢口駅であり、大学最寄り駅は一の鳥居駅とした。神戸電鉄の場合、都心部との接続が良い駅は新開地駅であり、大学最寄り駅は鈴蘭台駅とした。また、条件で与えた 5 つの駅は、上記 2 駅に隣接する駅である。

各路線での本タスク終了後、被験者にはクイズとアンケートに回答させる。本タスク実施前に被験者にはクイズ及びアンケートの内容を被験者に共有し、回答のために必要な情報の収集を同時に行わせる。タスクは VR システム、平面システムどちらを使用する場合も制限時間を 40 分と定める。この時間は、事前実験をもとにシステムに不慣れな被験者でも不動産物件を 1 件選択するのに十分となるよう設定している。

5.4 評価項目

評価は被験者の属性間の条件分布理解度と、VR システムのユーザビリティについて行う。被験者の属性間の条件分布理解度を評価するため、不動産情報に関するクイズ、アンケートを実施する。また、VR システムのユーザビリティを評価するため、VR システムに関するアンケートを実施する。

不動産情報に関するクイズでは、探索した不動産情報について、表 4 に示す設問を被験者に問う。クイズは選択肢の中から 1 つ被験者に選択させる。Q3、Q6 の設問は選択肢が 5 つであり、その他の設問は選択肢が 6 つである。また、全ての設問に分からないという選択肢が含まれる。選択肢が 5 つの設問は 1 つの選択肢が正解となる。選択肢が 6 つの設問は、実際の分布を最も正確に表している選択肢を正解、完全に正確ではないが一部実際の分布に合致する選択肢を準正解とする。例えば「指

表 6 VR システムに関するアンケート項目と平均スコア（1：そう思わない，5：そう思う）

質問番号	質問項目	平均スコア
Q1	全条件の物件を表示する不動産情報の俯瞰的表示機能は、閲覧して分かりやすかったか	3.50
Q2	全条件の物件を表示する不動産情報の俯瞰的表示機能により、不動産情報の分布などを理解することができたか	3.75
Q3	色分けや棒グラフでの属性間の条件ごとの分布を可視化する機能は、閲覧して分かりやすかったか	4.25
Q4	色分けや棒グラフでの属性間の条件ごとの分布を可視化する機能により、不動産情報の分布などを理解することができたか	3.50
Q5	色分けや棒グラフでの属性間の条件ごとの分布を可視化する機能は、絞り込みや物件の選択に役に立つ情報を提供していたか	3.75

表 7 使用システムごとの各回答種別における回答数

回答種別	VR システム	平面システム
正解	6	6
準正解	2	1
誤答	8	9
誤答のうち分からない	4	7

定した条件において選択肢の駅で最も物件数が多いのはどの駅か」という設問に対しては、最も不動産物件数が多い新開地駅が正解、次に多い湊川駅が準正解となる。不動産情報に関するアンケートでは選択した不動産物件について、不動産情報全体の中で比較した賃料などの特徴について自由記述形式で回答させる。

ユーザビリティの評価にはアンケートの他に System Usability Scale (SUS) [23] という指標を用いる。SUS は Brooke が提案した評価指標で、表 5 に示す 10 個の質問に 5 段階で評価する。

VR システムに関するアンケートでは、5 段階の選択式質問による定量的な評価と、自由記述形式の質問による定性的な評価を行う。VR システムに関するアンケートのうち、選択式質問の項目を表 6 に示す。自由記述形式の質問では、VR システムの俯瞰的表示機能と、属性間の条件ごとの分布可視化機能についての良し悪し、その他印象に残った点について記述させる。

5.5 結果

はじめに不動産情報に関するクイズの結果を述べる。クイズでは使用したシステムごとに、4 名ずつの被験者が各 4 問の設問に回答し、計 16 問ずつの回答を得た。使用システムごとの各回答種別における回答数を表 7 に示す。この表に示すように、使用するシステムによって正解率が変わらない結果となった。また、誤答のうち分からないという選択肢が選ばれた問題の数は、VR システムより平面システムの方が多という結果

となった。

次にユーザビリティについて結果を述べる。本実験での SUS の平均スコアは 49.375 であった。SUS の平均的なスコアは 68 とされている [24] ことから、VR システムはユーザビリティに課題が残るということが示された。

また、VR システムに関するアンケートで得られたスコアの平均値を表 6 に示す。この表に示すように多くの設問で高い評価を得た。特に属性間の条件分布可視化機能について、閲覧時の分かりやすさが高い評価を得ている。

次に VR システムに関する自由記述形式の質問の結果を述べる。VR システムの俯瞰的表示機能について、多くの被験者が条件ごとの物件数や分布が直感的に分かる点を長所として回答している。1 人の被験者は、物件数の多い条件が分かり、その条件を軸に探索を始めることができた点が、不動産情報探索の初心者にとって探し始める際の取っ掛かりになったと回答している。一方で多くの被験者が、欠点として表示する情報の視認性や理解の難しさに言及している。不動産情報パネルが近いため俯瞰して見づらいという回答や、不動産物件の自動的切り替えが速く、お気に入り登録などの操作に支障があったという回答が得られた。また、俯瞰的表示では各条件における不動産情報の有無は分かるが、不動産物件そのものの良し悪しは分かりづらいという回答が得られた。

属性間の条件分布の可視化機能について、多くの被験者が条件分布を直感的に理解可能な点を長所として評価している。また、ある最寄り駅における間取りごとの傾向を、条件の絞り込みをせずに比較できた点が長所として挙げられた。一方、短所として操作の複雑さや、一度に提示する情報量の少なさを指摘する回答が得られた。条件ごとに色を割り当てるのが面倒であるという回答や、可視化のための属性を選択する際の操作が直感的でないという回答が得られた。可視化する属性の選択は図 6 に示すメニュー上の、サブ属性の選択と書かれたボタンで行う。一方、各条件への色の割り当ては図 8 に示すように、表示した不動産情報への操作により行う。このような表示領域の異なる 2 種類のボタン操作が必要であったことが原因であると考えられる。また、可視化のための属性は同時に 2 つ以上選択できるほうが良いという改善点が指摘された。

6 まとめと今後の課題

本論文では、属性の分布を可視化し提示する VR インタフェースを提案した。提案するインタフェースでは、条件ごとに不動産情報をグループ分けし、各グループにおいて、もう一つの属性の条件ごとの不動産物件数割合を可視化して提示する。ユーザ実験により提案する VR システムと平面システムを比較し、被験者の属性間における条件ごとの分布理解度を調査した。その結果、ユーザの属性間の条件分布理解度には定量的な差が見られなかった。一方で被験者に実施したアンケートの結果から、提案システムが不動産情報の物件数分布や条件分布の理解を支援している可能性、さらに絞り込みなどの操作数を低減させる可能性が示された。また、ユーザビリティについての課題が明

らかになり、特に情報の表示位置をユーザから離すなどの改善点が示された。属性間の条件分布可視化機能については、同時に可視化する属性の数を増やすなどの改善点が示された。

今後の課題としてユーザビリティの改善が必要である。表示情報の大きさや位置を調節することや、操作の簡略化、操作ガイドの表示などに取り組む。また、実験で指摘を受けた可視化する属性の増加や、属性間の条件分布可視化手法について、他に有効な手法の検討に取り組む。

謝 辞

本研究は、JSPS 科研費 JP24K03228, JP25K03229, JP25K03228 の助成を受けたものです。また、本研究では、国立情報学研究所の IDR データセット提供サービスにより株式会社 LIFULL から提供を受けた「LIFULL HOME'S データセット」を利用しました。ここに記して謝意を表します。

文 献

- [1] Austin R. Ward and Rob Capra. Immersive Search: Using Virtual Reality to Examine How a Third Dimension Impacts the Searching Process. In *Proceedings of the 2020 International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1621–1624, 2020.
- [2] Austin Ward, Sandeep Avula, Hao-Fei Cheng, Sheikh Muhammad Sarwar, Vanessa Murdock, and Eugene Agichtein. Searching for Products in Virtual Reality: Understanding the Impact of Context and Result Presentation on User Experience. In *Proceedings of the 2023 International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2359–2363, 2023.
- [3] Austin Ward, Bogeum Choi, and Robert Capra. Eyes on Immersive Search: Eye-Tracking Study of Search Engine Result Pages in Immersive Virtual Environments. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, pp. 43–54, 2023.
- [4] Yumiho Takeuchi, Yoshiyuki Shoji, and Martin J. Dürst. Method of Loci in VR Web Search: Memory Retention Support by Organizing Search Results in a VR Room. In *Proceedings of the 2023 International Conference on Human-Computer Interaction*, pp. 373–391, 2023.
- [5] Daniele Giunchi, Stuart James, and Anthony Steed. 3D Sketching for Interactive Model Retrieval in Virtual Reality. In *Proceedings of the 2018 Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, pp. 1–12, 2018.
- [6] Maurice SchleuBinger. Information Retrieval Interfaces in Virtual Reality-A Scoping Review Focused on Current Generation Technology. *PLOS ONE*, pp. 1–24, 2021.
- [7] Tim Dwyer, Kim Marriott, Tobias Isenberg, Karsten Klein, Nathalie Riche, Falk Schreiber, Wolfgang Stuerzlinger, and Bruce H. Thomas. *Immersive Analytics: An Introduction*, pp. 1–23. Springer International Publishing, 2018.
- [8] Dane Coffey, Nicholas Malbraaten, Trung Le, Iman Borazjani, Fotis Sotiropoulos, and Daniel F. Keefe. Slice WIM: A Multi-Surface, Multi-Touch Interface for Overview+Detail Exploration of Volume Datasets in Virtual Reality. In *Symposium on Interactive 3D Graphics and Games*, pp. 191–198, 2011.
- [9] Yalong Yang, Maxime Cordeil, Johanna Beyer, Tim Dwyer, Kim Marriott, and Hanspeter Pfister. Embodied Navigation in Immersive Abstract Data Visualization: Is Overview+Detail or Zooming Better for 3D Scatterplots? *IEEE Transactions on Visualization and Computer Graphics*, pp. 1214–1224, 2021.
- [10] Gary Marchionini. Exploratory Search: From Finding to Understanding. *Communications of the ACM*, Vol. 49, No. 4, pp. 41–46, 2006.
- [11] Orland Hoerber. Design Principles for Exploratory Search Interfaces. In *Proceedings of the 2025 ACM SIGIR Conference on Human Information Interaction and Retrieval*, pp. 12–22, 2025.
- [12] Mohammad Najah Mehdi, Abdul Rahim Ahmad, and Roslan Ismail. A Visualization Technique to Support Searching Filtering. *MATEC Web Conf.*, pp. 1–9, 2019.
- [13] Alan Medlar, Jing Li, and Dorota Glowacka. Query Suggestions as Summarization in Exploratory Search. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, pp. 119–128, 2021.
- [14] Anita Crescenzi, Austin R. Ward, Yuan Li, and Rob Capra. Supporting Metacognition during Exploratory Search with the OrgBox. In *Proceedings of the 2021 International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1197–1207, 2021.
- [15] 吉田駿哉, 長谷川大, 佐久田博司. 探索的検索における Web ブラウザ履歴の可視化による知識の構造的把握. 情報処理学会インタラクティブ 2017 論文集, pp. 495–498, 2017.
- [16] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics*, pp. 649–658, 2016.
- [17] Mingzhao Li, Zhifeng Bao, Timos Sellis, Shi Yan, and Rui Zhang. HomeSeeker: A Visual Analytics System of Real Estate Data. *Journal of Visual Languages & Computing*, pp. 1–16, 2018.
- [18] Guodao Sun, RongHua Liang, FuLi Wu, and HuaMin Qu. A Web-based Visual Analytics System for Real Estate Data. *Science China Information Sciences*, pp. 1–13, 2013.
- [19] 諏訪博彦, 大坪淳, 中村優吾, 野口真史. IoT センシングによる賃貸物件快適度指標の構築と評価. 人工知能学会全国大会論文集, 2020.
- [20] Goran Milutinovic and Stefan Seipel. Visual GISwaps - An Interactive Visualization Framework for Geospatial Decision Making. In *Proceedings of the 2018 International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 236–243, 2018.
- [21] 湯本真樹. 条件検索とラフ集合の縮約ルールを利用した学生向け賃貸物件推薦システムの開発. 電気学会論文誌C (電子・情報・システム部門誌), pp. 740–748, 2013.
- [22] 中山裕紀, 大島裕明. 不動産探索 VR: 不動産情報探索のための VR インタフェース. 情報処理学会研究報告エンタテインメントコンピューティング, pp. 1–8, 2025.
- [23] John Brooke. SUS - A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, pp. 189–194, 1996.
- [24] Brandy Klug. An Overview of the System Usability Scale in Library Website and System Usability Testing. *Weave: Journal of Library User Experience*, 2017.