オブジェクト指向のイベントデータから特徴量を抽出およびエンコードするためのフレームワーク

要旨

従来のプロセスマイニング技術は、イベントデータをそのまま利用し、各イベントが正確に 1 つのオブジェクトと関連付けられている。オブジェクトはプロセスのインスタンス化を表している。オブジェクト指向のイベントデータは、複数のプロセスの相互作用を表現する複数のオブジェクトに関連するイベントを含む。従来のプロセスマイニング技術は、1 つのオブジェクトに関連するイベントを想定しているため、オブジェクト指向イベントデータには適用できない。従来のプロセスマイニング技術を利用するためには、オブジェクト指向のイベントデータを、1 つを除くすべてのオブジェクト参照を削除して平坦化する。この平坦化処理は非可逆的であり、平坦化されたデータから抽出される特徴量は不十分である。さらに、オブジェクト指向のイベントデータのグラフのような構造は、平坦化の際に失われる。本論文では、オブジェクト指向のイベントデータから特徴量を抽出し、符号化するための一般的なフレームワークを紹介する。私たちは、オブジェクト指向のイベントデータ上でネイティブに特徴量を計算し、正確な測定値を得ることができる。さらに、これらの特徴量に対して、表形式、逐次式、グラフベースの 3 つの符号化方式を提供する。表形式と逐次式はプロセスマイニングで多用されているが、グラフベースはオブジェクト指向のイベントデータの構造を保持した新しい手法である。3 つの符号化それぞれについて、可視化ユースケースと予測ユースケースの 6 つのユースケースを提供する。予測ユースケースでは、説明可能な AI を使用し、オブジェクト指向の特徴量とシーケンシャルとグラフベースの符号化の構造の両方が予測モデルにとって有用であることを示す。

1．導入

プロセスマイニング[1]は、プロセスによって生成されたイベントデータからデータ駆動型の洞察とアクションを生成するコンピュータサイエンスの一分野である。これらの洞察は、通常、プロセス発見、適合性チェック、およびエンハンスメントの 3 つのカテゴリに分類される。プロセス発見技術は、プロセス内のアクションの可能な経路を記述するプロセスモデルを作成する。適合性チェック技術は、プロセスモデルとイベントデータ間の対応関係を定量化・定性化する技術です。プロセス拡張技術は，イベントデータの特徴量を

符号化して入力とし、出力として洞察、予測、またはアクションを提供する。このような拡張技術には、プロセスのパフォーマンス分析［20,24]、予測［10,26,29]、類似プロセスの実行のクラスタリング［25]が含まれる。

Fig.1 挿入

一般に、プロセス拡張技術は、イベントデータの特徴量を表形式[18,9]または配列の集合[12,26,19]のどちらかの方法で符号化する。表形式では、各行が例えばイベントの特徴量に対応する。この表形式化は、例えば、回帰、決定木、フィードフォワードニューラルネットワークなどに利用される。しかし、各プロセスの実行(ケースともいう)は、時系列に並んだイベントである。したがって、イベントデータを表形式にまとめると、イベントデータの持つ順序性が失われる。この構造自体に意味があるため、逐次的な符号化方式が開発された[19]。これらの符号化は、各処理実行を特徴量の列として表現し，LSTM[26]などの逐次符号化データを考慮した予測モデルや処理実行の変種を可視化するために利用される。

従来のプロセスマイニングは、2 つの中心的な仮定に基づいて構築されています。各イベントは、正確に 1 つのオブジェクト(ケース)に関連付けられ、各オブジェクトは同じタイプである。各オブジェクトは、一連のイベントと関連付けられています。したがって、従来のイベントログは、均質に型付けされ、孤立したイベントのシーケンス のコレクションを記述します。これは、例えば、保険金請求の処理を分析する場合には、有効な仮定です。この例では、各オブジェクトは同じ型である保険金請求のインスタンス化を記述する。イベントは正確に 1 つの保険金請求に関連づけられる。しかし、現実の情報システムでは、しばしば別の絵が描かれる。イベントは、異なるタイプの複数のオブジェクトに関連している場合があります。

Fig.2 を挿入

複数の関連オブジェクトを持つイベントデータを生成する情報システムの最も顕著な例は、ERP システムである。このようなシステムにおけるオブジェクトは、例えば、注文、この注文の異なる項目、注文から現金化までのプロセスにおける請求書などに対応する。図 1 に示すようなオーダー処理プロセスの単純化された例を考えてみよう。イベントは、注文、品目、またはその両方のタイプのオブジェクトに関連する可能性があります。複数のオブジェクトを持つイベントは、複数の先行イベントを持つ可能性がある。したがって、オブジェクト指向イベントログ(OCEL)の構造は、従来のプロセスマイニングで想定されているようなシーケンシャルな構造ではなく、グラフに類似している。
OCEL と従来のプロセス拡張技術の間のこのギャップは、現在、イベントログの平坦化

[2]、すなわち、均質で連続した構造を強制することによって、OCEL を従来のイベントログ形式にマッピングすることによって埋められています。これには 2 つの段階があります。ケース概念を選択し、その概念の複数のオブジェクトを持つイベントを複製する。このケース概念に含まれないオブジェクトはすべて破棄される。図 1 のイベントログを 3 つの異なるケース概念で平坦化したのが図 2 である。最初の 2 つは、1 つのオブジェクトタイプのケース概念である[2]。

Fig.3 を挿入

第 3 の事例概念は、オーダーとアイテムが共存する複合事例概念、すなわち、オーダーとそれに対応するすべてのアイテムの事例概念である。平坦化されたイベントデータは、従来のプロセス拡張技術の入力として使用することができる。
しかし、フラット化は、オブジェクト指向のイベントログの情報を操作するものである。フラット化に関連する問題として、欠損(イベントが消える) [3]、収束(イベントが重複する) [2]、発散(直行関係を誤解させる) [28,2]がある。ダイバージェンスについて例を挙げて説明する。イベントログを平坦化するために、オーダーとアイテムの複合ケース概念を使うことができる(図 2 ケース概念：オーダーとアイテム)。イベントを通じて関連するすべてのオーダーとアイテムは、o1、i1、i2、 o2、i3 という一つの複合オブジェクトを形成する。これらのオブジェクトのイベントは 1 つのシーケンスにフラット化され、不正確な優先順位制約が導入される。例えば、品物を取るというイベント e3 と e4 は順番に並べられ、両者の間に何らかの順序があることが示される。しかし、元のイベントデータでは、この 2 つのピッキングイベントは独立であることがわかる。ピッキング商品と支払い順の関係も同様である。オブジェクト指向のイベントデータでは、優先順位の制約はない。オブジェクト指向のイベントデータは優先順位の制約を示さないが、逐次表現では強制される。

イベントの欠落、重複、誤った優先順位制約のために、多くの特徴量が正しくない結果をもたらす(第 4 節を参照)。さらに、これらの特徴量から構築される表形式または逐次的な符号化は、イベントデータのグラフ構造を保持しないため、重要な構造情報が削除される。したがって、OCEL のための特徴量を正確に抽出し、符号化すること はできない。

Table 1 を挿入

この問題を解決するためには、オブジェクト指向のイベントデータに対してネイティブに特徴量を計算し、イベントログの実際の構造を保持したままグラフベースで符号化するアプローチが必要である。本論文では、オブジェクト指向のイベントデータ(図 3 参照)に対して特徴量を抽出し、符号化するための一般的なフレームワークを紹介し、以下の 2 つ

の貢献を行う。1) de Leoni et al [18]のフレームワークで導入された特徴量の計算を、オブジェクト指向の設定に変換し、正確な尺度を提供する。2) 抽出された特徴量を、異なるアルゴリズムや手法で表現するために、表形式、逐次式、グラフベースの 3 種類の符号化を提供する。特徴量と符号化を用いて、6 つのユースケースを提供する。これらのユースケースは、多くの異なるタスクに対する我々のフレームワークの一般性を示している。各符号化に対して、1 つの可視化ユースケースと 1 つの予測ユースケースを使用する。予測ユースケースでは、説明可能な AI と SHAP 値[21]を活用し、予測モデルによって符号化の異なる特徴量と構造がどのように利用されるかを描写している。これらの貢献は、新しいアルゴリズム、新しい可視化、新しい機械学習モデル、より正確な予測などのための基礎として使用することができます。

本論文は以下のような構成になっている。まず、Sec.2 で特徴量抽出と符号化に関する関連研究を説明する。Sec.3 では、オブジェクト指向のイベントデータとプロセス実行について紹介する。第 4 項では、オブジェクト指向のイベントデータに対するネイティブな特徴量計算の概要を説明する。第 5 章では、オブジェクト指向の特徴量に対する 3 つの符号化方式を定義する。第 6 章では、特徴量とその符号化方式に関する 6 つのユースケースを示す。第 7 章では、本論文の結論を述べる。

2 .関連作業

プロセス性能分析、予測的プロセス監視、トレースクラスタリングなど、多くのプロセス拡張技術が文献に存在する[1]。これらの技術は、イベントログから抽出された符号化された特徴量を入力として用いる。表 1(a)は、特徴量抽出(1.OCEL を用いた特徴量抽出、2.イベントログの平坦化)と符号化(1.表形式、2.逐次、3.グラフ)の手法を変えた 3 種類のカテゴリを代表例とともに示したものである。まず、P1 は、平坦化されたイベントログから抽出された特徴量を表形式に符号化する技術を表している。例えば、van Dongen ら[9]は、イベントログを特徴量と結果の組に変換して表形式化し、ノンパラメトリック回帰を用いて残り時間を予測する手法を用いている。

また、[13]では、イベントログを表形式に符号化し、さらに、リソースの有無などのコンテキストに関する特徴量を追加して、処理時間を予測することが示されている。第二に、P2 の技術は、平坦化されたイベントログに基づく特徴量も使用するが、シーケンシャルなフォーマットとしてエンコードされる。Leontjeva ら[19]は、進行中のケースの結果を予測するために、イベントログをシーケンスにエンコードする複雑なシーケンス符号化を提案している。また、Evermann ら[12]は、進行中の案件の次の活動を予測するために、制御フロー特徴量を埋め込み技術で符号化し、Tax ら[26]は、ワンホット符号化を用いる。最後に、P3 は、フラット化されたイベントログから抽出された特徴量をグラフ化す

る技術である。Philipp ら[23]は、イベントログを、各ノードがアクティビティを表し、各エッジがアクティビティ間の関係を示すグラフに符号化する。このグラフは、プロセスの結果を予測するために、グラフニューラルネットワーク(GNN)を学習するために使用される。Venugopal ら[27]は[23]を拡張し、ノードに時間的特徴量を注釈している。彼らは、GNN を使用して、イベントの次のアクティビティと次のタイムスタンプを予測する。
Harl ら[16]はノードを活動として表現する代わりに、活動のワンホット符号化を用いてノードを表現し、関連性スコアに基づく説明可能性を提供するゲートドグラフニューラルネットワークを展開する。

さらに、異なる特徴量抽出・符号化を用いたプロセス拡張技術の開発を支援するために、いくつかのフレームワークが提案されている(表 1(b)参照)。まず、F1 の De Leoni [18] は、フラット化されたイベントログを用いて特徴量を計算し、テーブルにエンコードするフレームワークを提案している。次に、F2 の Becker ら [7]と Di Francescomarino ら [14] は、抽出した特徴量を逐次符号化する技術の枠組みを提案し ている。我々の知る限り、グラフ符号化をサポートするフレームワークは存在しない。
イベントログの平坦化には誤解を招く特徴量の抽出に限界があるにもかかわらず、OCEL に基づく特徴量を用いたプロセス拡張技術の開発は行われていない。本研究では、オブジェクト指向のプロセス拡張手法の開発を促進することを目的として、OCEL に基づく特徴量の抽出とエンコードのためのフレームワークを提供する。提案するフレームワークは、表形式、逐次形式、グラフ形式といった既存のすべての符号化形式をサポートし、異なるアルゴリズムや手法に対応する。

3.オブジェクト指向イベントデータ

集合 X が与えられたとき、冪集合 P(X)は可能なすべての部分集合の集合を示す。
列[σ:{1, ... n}→X の長さ len(σ)=n]は、X の要素に順序を割り当てているものである。
数列を $\sigma = \langle x_1, \dots, x_n \rangle$ とし、X上のすべての数列の集合をX^*とする。

続きあり

4.オブジェクト指向特徴量

ここでは、OCEL をフラットにしてプロセス拡張技術を適用した場合の問題を取り上げる。プロセスエンハンスメント技術を適用するために操作され、平坦化されたイベントデータに対して、特徴量が計算される。平坦化されたイベントデータに対して計算される。

そのため、特徴量は不正確である可能性がある。我々は、de Leoni らの機械学習フレームワークで導入された特徴量をオブジェクトセントリックに適応させることを提案する。de Leoni ら[18]の機械学習フレームワークで導入された特徴量のオブジェクト中心への適応を提案する。我々は、OCEL 上でこれらの特徴量をネイティブに計算する。さらに、我々はオブジェクト指向型プロセスマイニングに関する文献で最近紹介されたいくつかの新しい特徴量を提供する。特徴量とは、一般的に、あるイベントに対して算出されるものである。それは、単一のイベント、そのプロセス実行との関係、またはシステム全体に対する尺度を記述することができる。

定義 5(特徴量)

従来の特徴量計算を適応させる主な必要性は、オブジェクト中心のイベントデータとの 2 つの主な違いから生じている。オブジェクト指向型と従来のイベントデータとの主な相違点である。まず、1 つ目、各イベントは、オブジェクトごとに 1 つずつ、複数のプリデクサー/サクセサーを持つことができる。第二に、各イベントは異なるタイプの複数のオブジェクトを持つ可能性がある。前後の動作に依存する特徴量の計算は、グラフ構造に合わせる必要がある。最も分かりやすい例は先行アクティビティである。従来の特徴量抽出では、各イベントに先行するアクティビティは 1 つだけであった。オブジェクト指向の特徴量抽出では、各オブジェクトに対して複数の先行アクティビティが存在する。また、グラフ構造とオブジェクトの多重性により、グラフ構造とオブジェクト(型)の関連性を利用した新しい特徴量を定義することができる。前のイベント(すなわち、実行中の考慮イベントの前に起こったすべてのイベント)と後のイベントは、時間ベース(イベントのタイムスタンプを使用)とパスベース(グラフのパス情報を使用)の 2 つの方法で適応させることができます。我々は単純な時間ベースの適応を採用しています。しかしながら、グラフを用いた適応は、新たな研究の方向性を与えるかもしれない。

Fig.4 を挿入

図 4 に、de Leoni ら[18]のフレームワークをオブジェクト中心 に適用して収集した特徴量と、最近の文献[3,22]で紹介された 特徴量の概観を示す。de Leoni らと同様に、異なる視点によって特徴量をグループ化している。異なる視点とは、制御フロー、データフロー、リソース、パフォーマンス、オブジェクトである。ここでは、異なる視点と、それらをオブジェクト指向の設定に適用するために必要な適応について説明する。

制御フローの観点からの主な改良点は、逐次的な制御フローからグラフ的な制御フローへの切り替えに関わるものである。複数の先行アクティビティ(C2)だけでなく、複数の現在アクティビティ(C1)(現在の実行グラフの終点)も可能である。前後のアクティビティ

(C3、C4)については、単純な時間ベースの適応を用いる。

Table2 を挿入

データフローの観点では、先行する特性値(D2)に対して若干の適応が必要である。先行する値が複数存在する可能性があるため、これらを集計する必要がある。以前の特性値(D1)は時間軸で適応され、特性値(D3)は適応が不要である。

リソース視点の特徴量は、現在のリソースの作業負荷（C1）やシステム全体の作業負荷（C2）など、主にシステム全体の測定に関係しています。そのため、この視点はオブジェクト指向型への移行にもほとんど影響を受けず、オブジェクト指向型に移行しても、ほとんど影響を受けません。今後は、イベントごとのリソースの多重度から得られる新たな特徴量を研究することも考えられます。

パフォーマンスの観点からは、最近、新しいオブジェクト指向型の特徴量について研究されている[22]。イベントは複数の先行イベントを持つため、オブジェクト間の同期時間（P5)、オブジェクトタイプのプーリング時間（P7)、イベント前のオブジェクトタイプ間のラグ（P8）を表現するいくつかの特徴量によって、確立したパフォーマンス指標を拡張することができます。

最後に、オブジェクトに関する新しい特徴量的な視点が開かれる。ブジェクト中心のペトリネットの発見を紹介した論文[3]では、オブジェクトの視点での基本的なオブジェクトの視点の基本的な特徴量を紹介している。例えば、イベントのオブジェクト数(O5)、イベントの特定タイプのオブジェクト数(O6)、現在のシステムの総オブジェクト数(O1)などが挙げられる。このような観点からの追加特徴量の検討、例えば、オブジェクトグラフのグラフメトリクスによるオブジェクト間の関係の定量化も興味深い研究方向となる可能性がある。

5.特徴量の符号化

Fig5 を挿入

ここでは、オブジェクト指向のイベントデータのグラフ的構造を表現する特徴量の符号化がないことに取り組む。現在使われている表形式と逐次式の符号化をグラフベースのものに拡張し、3 つの符号化すべてを正式に紹介する。各エンコーディングの正式な定義とともに、一般的な使用例、利点、欠点、および図 1 の実行例の続きを紹介します。抽出された特徴量の例として、前のオブジェクトの数（O2)、同期時間（P5)、残り時間（P3）を挙げる。この例の実行抽出は、連結成分抽出EX_{comp}である。表形式符号化(Tabular Encoding)は、回帰分析、クラスタリング、さまざまなデータマイニングなど、多くのユ

ースケースで使用されるデータポイントの一般的な表現である。

定義 6(表形式符号化(Tabular Encoding))

定義 7(逐次符号化(Sequential Encoding))

Table 3 を挿入

図 5 b)に運用例の逐次符号化を示す。プロセス実行のためのイベントは、イベントの完全なタイムスタンプに従って並べられます。その結果得られたシーケンスには、イベントごとに異なる特徴量が付与される。この符号化は、イベントの適時な順序を尊重します。しかし、これはイベントログの真の優先順位の制約を尊重しません。すべてのイベントを 1 つのシーケンスに統合することで、いくつかのイベントのペアは、イベントログで示されなかった優先順位関係になる(Sec.1 参照)。特徴量のグラフ化は、広範な可視化、グラフアルゴリズムの適用、またはグラフニューラルネットワークの活用のために使用することができます[30]。

定義 8(グラフ符号化(Graph Encoding))

図 5 c）に実行例に対するグラフベースの特徴量符号化の例を示します。各プロセスの実行はグラフと関連付けられています。グラフの各ノードがイベントの特徴量を表します。

6 .活用事例

本章では、6 つの活用事例を提示し、本フレームワークを評価する。このアプローチでは、2 つの評価目標を掲げている。まず、フレームワークが適用可能な一般的なプロセスマイニングタスクのコレクションを提供することで、フレームワークの一般性を示すことを目的としている。第二に、活用事例における特徴量と符号化の有効性を示すことを目的としている。ここ数年、説明可能な AI は、予測プロセスの監視を透明化するためにますます採用されるようになっている[15,17]。SHAP [21]値を用いることで、特徴量の重要性とともに、逐次符号化およびグラフに基づく符号化の構造的重要性を定量化することが可能である。

Fig.6 を挿入

活用事例は、3 つの可視化活用事例と 3 つの予測活用事例の 2 つに分けられる。OCEL として、実際のローン申し込みのイベントログ[8]を使用する。イベントは、アプリケーションと複数のローンオファーにオブジェクトとして関連付けることができる。表形式、逐次式、グラフの符号化を使用し、可視化の活用事例を通してプロセスのインサイトを得ることができる。予測活用事例は、イベントのプロセス実行の残り時間（P3）を、回帰（表形式)、LSTM ニューラルネットワーク（逐次)、GNN（グラフベース）という異なる符号化の 3 種類の手法で予測することを目的としている。各エンコーディングに同じ特徴量を使用する。先行アクティビティ(C2)、平均先行要求量(D1)、経過時間(P2)、先行オファー数(O3)である。比較のため、各モデルで同じイベントを 0.7/0.3 の割合で訓練/テストに分けて使用しています。トレーニングセットの 20%を検証セットとして確保した。パフォーマンスは、正規化されたターゲット変数の平均絶対誤差（MAE）を用いて評価される。さらに、トレーニングセットの平均残存時間を予測することによって達成されるベースライン MAE を提供する。表 3 はその結果をまとめたものである。本フレームワークの実装をオープンソースの Python で提供する(https://github.com/ocpm/ocpa)。このフレームワークは、新しい機能や適応的なアルゴリズムで拡張することが可能である。

6 .1.表形式符号化(Tabular Encoding)

可視化
イベントログを、1 週間分のイベントを含む後述のサブログに分割した。
各サブログについて、平均要求額（D3)、イベントごとのオファー数（O6）を抽出する。その結果、図 6 に示すような時系列が得られた。時間の経過とともに要求量が増えるなど、プロセスのダイナミクスを観察することができる。さらに、オファー数は、いくつかの短いスパイクを除いて、安定していることが観察されます。