

FAにおけるエッジアプリケーションフレームワークの実装と評価

境 将輝[†] 中島 毅[†]
 芝浦工業大学[‡]

高橋 和也[‡]
 Edgexcross コンソーシアム[‡]

1. 研究背景

FA (Factory Automation) のIoT化が急速に進み、製造機器に近いエッジ領域においてリアルタイム性の高い分散処理を行うエッジアプリケーション(以下 EAP)技術が注目されている。従来、生産ラインから取得したデータはオフライン上で分析され、その分析結果を生産ラインのオペレーションに反映させることで、生産性向上のために利用されてきた。この改善サイクルの効率を上げるために、オフライン分析の結果を直接 EAP に反映することで、リアルタイムに機器を監視・制御することが求められている。

しかし、オフライン分析には多くの手法があること、製造現場のデータを取得する方法が製造機器ごとに異なることから、それらの差異によって発生する作業負担を軽減させるような EAP の開発支援が必要である[1]。分析結果を反映するためには、推論モデルの利用が必要となり、それにはモデルの作成だけでなく、機器から取得したデータを分析に活用するためのデータ加工処理(前処理)に関する負担軽減が求められている。

2. 先行研究と課題

先行研究として EAP の開発支援フレームワークがある[2]。EAP は、生産現場における生産手順の変更や製造機器の変更によって、中身の更新を行う必要があり、開発時の他、そのような更新時にもプログラミングが必要となる。フレームワークはこれらの状況下でのプログラミングレス化を目的としている。

図 1 はフレームワークの構想を示す図である。フレームワークはオフライン分析での結果を用いることで、リアルタイムで取得したデータに対して推論を行い、推論結果に応じて機器の制御を行う。オフライン分析での結果をリアルタイムでの推論に反映させるには以下の3つのモデルを使用する。

- ① データモデル: 統合データ処理層から出力されたデータ項目の定義。主な項目としてデータの型やデータ名などがある。オフライン上、オンライン上のどちらでも同じデータ名でアクセスできることを前提とする。
- ② 前処理記述: 収集したデータを推論モデルが必要とする説明変数に変換するための前処理方法の定義
- ③ データマイニングモデル: データ分析結果を表現する推論モデル。

作成されたモデルに基づき、EAP は①製造機器からデータの取得、②取得したデータに対して前処理の実行、③前処理によって

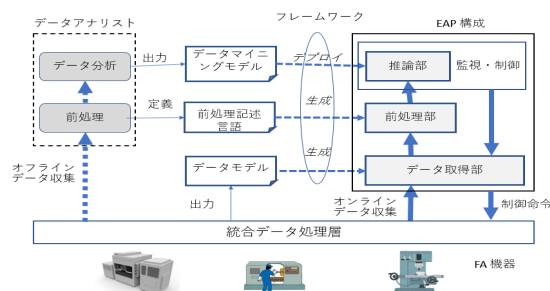


図 1 EAP フレームワーク構想図

作成されたデータを用いた推論の実行の3つの動作を制御する。

EAP フレームワークでは、モデルによって EAP の挙動を管理しているため、EAP の開発・更新時においてプログラミングレス化を達成しているが、1 節で述べたデータの前処理に関して、以下2つの課題が存在する。

- (1) 搭載している前処理の数が少ない

先行研究のフレームワークには、前処理を行う際、平均値(average)、最大値(max)、最小値(min)、標準偏差(sd)、分散(var)、最大値と最小値の差(RangeSpan)を求める機能しか備えておらず FA 分野での多様な分析に対応することが難しい。

- (2) 処理同士の組合せが行えない

前処理には様々な種類が存在し、その前処理を組み合わせることで、推論モデルに入力する説明変数を作成することが多い。しかし先行研究のフレームワークでは、単体での処理しか行うことができない。そのため、複数の処理を柔軟に組合せるような仕組みが必要である。

3. 研究目的

本研究は、従来のフレームワークの前処理に関する課題を解決する手法の提案を目的とする。従来の前処理記述に新たな前処理記述を加えることで、フレームワークの前処理の網羅性を高める。

4. 前処理の分類

一般的に行われる前処理には、データ構造を対象とした前処理とデータ内容を対象とした前処理の2種類がある[3]。この2種類の前処理をさらに細分化することによって、どのような前処理を行うときに、どのような表現を使うべきなのかを明確にする。

4.1 データ構造を対象とした前処理

データ構造を対象とした前処理では、製造機器から取得したデータの中から、実際に使用されるデータ項目の抽出やデータの値を損なわずにデータ量を削減するデータ集約などを行う。

データの集約では、従来研究で実装されているような統計量を求める処理の他に、ある項目のデータの合計値、中央値、最頻値など様々な数値を求める処理がある。

「Implementation and Evaluation of Edge Application Framework in FA」

[†] 「Masaki Sakai・Shibaura Institute of Technology」

[‡] 「Tsuyoshi Nakajima・Shibaura Institute of Technology」

[‡] 「Kazuwa Takahashi・Edgexcross consortium」

4.2 データ内容を対象とした前処理

データ内容を対象とした前処理では、取得した数値データに対して変換式を用いて数値・単位の変換を行うスケール変換や、取得したデータに欠損値が存在した場合に行う欠損値補完、カテゴリ型の変数を数値変換する離散化などがある。

5. 前処理記述について

5.1 前処理記述実現に対する要求事項

提案する前処理記述では、以下の2つを満たしている必要がある。

(1) 新しい処理を容易に組み込めること

1節より、データ分析には多種多様な方法が存在する。それらに対応するため、フレームワークが提供する前処理だけでなく、新しい前処理手法を柔軟に追加できるような仕組みが必要である。また、新しい手法の追加を容易にする際、閾値のみが違い、処理内容が同じ処理の追加など、意味のない追加が起こらないようにシンプルかつ理解しやすい構成であることも重要である。

(2) 様々な前処理を組み合わせることができること

2節より、1つの前処理を単体で扱うのではなくそれらの組合せを柔軟に行えるような仕組みが必要である。前処理の組合せには、ある単純な前処理の接続だけでなく、前処理後の複数のデータセット間の演算処理などを行う手法が存在する。

5.2 提案する前処理記述

上記の課題を解決するために、①前処理の抽象化・定義を行い、②実際のデータに対して定義した前処理を当てはめるという手法を採用。前処理定義では処理内容の定義を行い、説明変数の作成では、行う前処理と用いるデータを指定して前処理の方法を決定する。記述の方式はXML形式で、必要な情報(処理対象・処理内容)をタグで囲む構造となっている。

提案する前処理記述を用いた例を以下に示す。使用するデータにはFisher's Iris Dataset[4]を用いた。図2は考案した前処理記述の例である。記述は、“ある2つのデータ群に対して、それぞれ閾値が異なる二値化を行い作成したデータ同士の乗算を行う”という処理を表している。以下は例を用いた①、②についての詳細である。

①前処理内容定義では対象データがある閾値を境に二値化される処理と対象データ2つの乗算を行う処理を定義している。二値化処理では対象データと閾値の抽象化(data, threshold), 乗算処理では対象データの抽象化(data1, data2)を行っている。説明変数作成で当てはめる前処理の識別を行うために、定義した前処理にはprocessingタグのname属性で名前を設定する。定義した内容をそのまま使用できるので記述者は新しい処理を容易に組み込むことができる。これにより、(1)の問題を解決している。また処理に使うパラメータを抽象化することにより、無駄な処理の追加も防ぐことができる。

②説明変数作成では、定義した前処理を適用する。processingタグのprocessing属性で定義した処理内容の名前を入力し、data, thresholdタグには対象データ名、閾値を入力する。現状、提案する記述方式では対象のデータ項目の全てのデータセットを範囲としている。また処理によって作成されたデータに対しても名前を定義する。これは前処理の組み合わせを行う際、どの処理を施したデータを用いるのかを識別するために使用する。例では、1つ目の二値化処理から作成されたデータを“processing1”と定義し2つ目の二値化処理から作成されたデータを“processing2”と定義している。これによって、乗算を行う際に定義し

```
<!-- ① 前処理内容定義 -->
<!-- 二値化定義 -->
<processing name = "binarization">
  <category1 processed_data = 0>
    <data <= threshold</category1>
  <category2 processed_data = 1>
    <data > threshold</category2>
</processing>
<!-- 乗算定義 -->
<processing name = "*">
  <data1 = data1
</processing>

<!-- ② 説明変数作成 -->
<variable name = "variable1">
  <processing name = "processing_data" processing = "*">
    <data1>processing1</data1>
    <data2>processing2</data2>
    <processing name = "processing1" processing = "binarization">
      <data>Sepallength</data>
      <threshold>5.0</threshold>
    </processing>
    <processing name = "processing2" processing = "binarization">
      <data>Sepallength</data>
      <threshold>2.0</threshold>
    </processing>
  </processing>
</variable>
```

図2 前処理記述例

た2つの名前を、乗算を行う対象データタグ(data1, data2)に当てはめることで、“閾値が異なる二値化を行い、作成した変数同士の乗算を行う”という前処理の組み合わせを実現している。処理内容の定義と説明変数の作成を分けることで、それぞれをシンプルに表現している。組合せの際、記述者が用いる処理とその処理に必要なパラメータを記述するだけで容易に処理を組み合わせることで(2)の問題を解決している。

提案手法では、自分で処理内容を定義するため、分析者が実現したい処理も新たに追加できる。二値化のように“条件に応じて数値を変換する処理”は他の離散化手法にも応用できると考える。

6. まとめと今後の課題

本研究では、従来のEAPフレームワークの前処理機能に着目した改善提案を行った。提案する前処理記述では、従来の課題である前処理の汎用性の問題を解決している。

しかし、提案する前処理記述ではデータ項目の選択が行えるがデータ範囲の選択が行えない。データは日時型や時系列での管理を行うこともあるため、様々な方法でのデータ範囲の選択を行える仕組みが必要である。また前処理には集約や離散化など様々な処理があり前者は複数のデータから単数データを出力するが、後者は複数のデータを出力する。そのため処理によって得られるデータのスケールが異なり、その点においても識別を行う必要があるが現段階ではその仕組みを持っていない。これらを以下に簡単に記述上で表現できるかが今後の課題となる。

参考文献

- [1] 製造業におけるIoTの活用例と将来像：e-Factoryを例として (特集 IoT がもたらす製造業の新展開) オペレーションズ・リサーチ Communications of the Operations Research Society of Japan: 経営の科学, 2018(柳生理子 高橋 雄一 大谷 治之 著)
- [2] Sakai, Masaki, Tsuyoshi Nakajima, and Kazuya Takahashi. "A Proposal for Edge Application Framework for Smart Factory." 2020 IEEE 44th Annual Computers, Software, and Applications Conference. IEEE, 2020.
- [3] 本橋 智光(2018) 前処理大全[データ分析のためのSQL/R/Python実践テクニック] 株式会社ホクソエム監修, 技術評論社
- [4] The Iris Dataset-scikit-learn 0.24.0 documentation https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html