

意思決定のためのビジュアルプログラミングによる データ分析を支援するWebアプリケーションの開発

富山県立大学工学部電子・情報工学科
2055017 沼田 賢一

情報基盤工学講座
指導教員：奥原浩之

1 はじめに

近年、企業などでは世間に溢れる様々な情報を収集し、ビックデータと呼ばれる非常に巨大で複雑なデータの集合として扱うことが増えてきている。そして、ビックデータを扱うには様々な処理や解析によって情報を取捨選択し、自分たちに必要な形で保管する必要がある。

また、ビックデータの情報量は膨大で、人の手で全て解析するのは困難であるため、一般にプログラミング言語を用いて機械に処理させるのが一般的である。よって、ビックデータを扱うためにはプログラミングの知識や技術が必要不可欠で、プログラミングに触れたことがない人には扱いづらいものになっている[1]。

そこで、プログラミング初心者でも扱いやすいビジュアルプログラミング言語を使い、ブロック1つ1つに処理を対応させることでブロックを並べるだけで処理できるプログラムを作成することができるようになる。プログラミングができない人でもビックデータを扱うことができると考えられる。

キーワード：Webサービス、データ解析、ビジュアルプログラミング言語、意思決定支援

2 デジタルトランスフォーメーション

2.1 Society 5.0 と web サービス

Society 5.0とは、サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立する人間中心の社会のことである。

本研究では、サイバー空間上でwebサービスとしてビッグデータを扱うことができるblocklyを提供し、ユーザーがフィジカル空間上でビックデータを扱えるようにすることでSociety 5.0を実現する。また、このようなITを利用した人々の生活をあらゆる面でより良い方向に変化させることをデジタルトランスフォーメーション[2]という。



図 1: Society5.0

2.2 サーバサイドプログラミングの技術

Web ブラウザは、HTTP を使って Web サーバーと通信する。このとき、ブラウザの操作によってブラウザからサーバーへ HTTP リクエストが送信される。また、Web

サーバーではクライアントからのリクエストを待ち、受信したら処理をして HTTP レスポンスをブラウザに返す。

本研究では、HTTP 通信を行ったあとに動的サイトとして、ユーザーからの入力に対しての処理を、非同期通信を使って Web サーバ上でプログラムの実行が要求され、結果をウェブブラウザに対して送信するシステムを開発する。クライアントサイドコードに使うことができる言語は、HTML,CSS,JavaScript であるが、サーバー側による処理を挟むことでそれ以外の言語(Perl,PHP,Python,Rubyなど)を使う事ができるようになる。他の言語を扱うことができるため、クライアントサイドにはないライブラリを使うことができる。

3 ビジュアルプログラミング

3.1 種々のビジュアルプログラミング

ビジュアルプログラミングとは、一般的に想像されるプログラミング言語ごとのルールに沿ってキーボード入力でテキストのプログラムコードを打つテキストプログラミングと違い、プログラムコードの要素ごとに視覚的なパートとして分解されたものをマウスによる操作で組み合わせてプログラムコードをつくるものである。テキストプログラムと比較して、ビジュアルプログラミングは子供にもわかりやすくプログラムを作ることができる。

すでにあるビジュアルプログラミングとして、機械学習（人工知能・AI）を使って課題を解決するクラウドサービスの MAGELLAN BLOCKS (BLOCKS) や教育用作られ様々なアプリケーションに使われている Blockly[4]などがある。また、blockly をベースにして作られたサービスとして Scratch や MakeCode が存在する。



図 2: Scratch

3.2 blockly の仕組み

Blocklyとは、Google が提供しているビジュアルプログラミング言語のオープンソースライブラリであり、簡単な操作で自分だけのビジュアルプログラミング言語によるプログラムコードを作ることができる。またカスタムブロックという機能があり、もともとあるブロックの他に、好きな処理をさせるブロックをユーザーが作成することができる。カスタムブロックの構成としては、見た目の定義に関するものとカスタムブロック自体の動作の定義を決めたも

のによって作られます。見た目の定義としては、ブロックの色やパズルのような前後や横の接続を定めたり、データの接続や入力に対しての変数などの型を決める。また動作に関する定義としては、前後の接続や入力に対しての処理をプログラムして決める。

本研究では、blockly でビックデータを処理することができるようカスタムブロックの機能を使ってオリジナルのブロックを作成する。

4 提案手法

本研究では、blockly でビックデータを解析できるオリジナルのブロックを作成する。2.2 にあるように、クライアントサイドのみでは、限られたプログラミング言語とそのライブラリしか使えないためサーバーサイドでの処理をブロックを通して行う。そのため、Asynchronous Javascript And XML(Ajax)[3] を使ってサーバーと通信を行う。また、Ajax による通信から先のバックエンドでは、Common Gateway Interface(CGI) と呼ばれるプログラムを使って、クライアントから送られたデータを処理して、結果をクライアントに返す。CGI は、python で記述し sklearn と呼ばれる機械学習ライブラリを本研究では使用した。

処理する対象となる入力データは、データの入力形式をあらかじめ統一し、拡張子は csv とすることにした。入力形式としては、セルの1行1列目に入力データ数、1行2列目に説明変数、2行2列目に目的変数、3列目にラベル名、4列目に基本データ型、そして5列目以降に3,4列目に対応する入力データが入るような形とした。

また、blockly にはもともとファイルを読み込むブロックがないため、ビックデータを処理するブロック以外にファイルを読み込むブロックも作成することにした。今回作るのは、入力形式が統一されてある csv ファイルの読み込みブロックと線形回帰の計算を行うブロックと、出力結果を csv 形式でダウンロードできるブロックを作成する。線形回帰とは、回帰分析と呼ばれる教師あり学習の1種で、1つ以上の特徴量と連続値の目的変数との関係をモデルとして表現することができる。

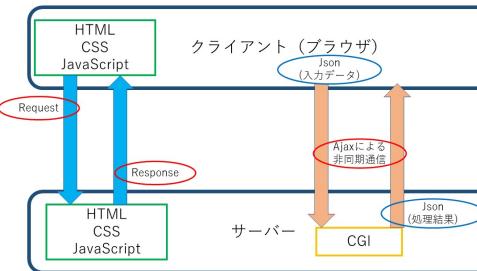


図 3: 開発システムのデータの流れ

5 実験結果ならびに考察

作成したファイルを読み込むブロックは図4のような見た目のブロックで作成した。

*** ファイルを選択してください *** を読み込み

図 4: ファイルを読み込むブロック

このブロックの使い方としては、**ファイルを選択してください**の枠内をマウスでクリックすることで、ポップアップでエキスプローラーが開き、そこから入力データとなる csv ファイルを選択すると、入力データを json に置き換えてこのブロック自体に内容を保持させることができます。

また、線形回帰を行うブロックは、図5のような見た目のブロックで作成した。



図 5: 線形回帰のブロック

このブロックの使い方としては、ブロックの空いたところに入力データとなる json が含まれるブロックを入れることで実行時に線形回帰の計算を行なわれ、回帰係数、切片、決定係数といった結果が帰ってくるものとなっている。ただし、今回の線形回帰のブロックは単回帰分析のみで、重回帰分析のときのブロックは作成していない。また、入力データは提案手法で示したようにあらかじめ統一されたデータ形式である必要がある。

最後に、入力データに対しての処理後の出力結果を csv 形式でダウンロードできようのブロックは図6のような見た目で作成した。



図 6: 出力結果のダウンロードができるブロック

このブロックの使い方としては、ブロックの空いたところを入力として、プログラムの実行時に自動で入力部分の処理結果を csv に変換し、そのファイルのダウンロードが開始されるものとなっている。

6 おわりに

本研究ではビジュアルプログラミング言語によるビックデータ解析システムの開発をした。クライアントサイドに存在しないライブラリを Ajax 通信を使ってサーバーサイドから必要なライブラリとそれを実行する CGI プログラムを呼び出すことで blockly において線形回帰の計算の結果をビジュアルプログラム上で得ることができた。

今後の課題は、ビックデータを分析する手法を調べ、そのブロックを作成していく。また、今回作った線形回帰のブロックを編集してオプションで単回帰分析が重回帰分析をブロックで選択できるようにする。

参考文献

- [1] 野口孝文, 千田和範, 稲守栄. (2015), 初心者から上級者までシームレスにプログラミングを学ぶことができる持続可能な学習環境の構築. 教育システム情報学会誌, 32(1), 59-70.
- [2] 加藤智之, 越島一郎, 梅田富雄 “デジタルトランスフォーメーションに関するP2Mの展開”, Journal of International Association of P2M Vol.13 No.1, pp.107-120, 2018.
- [3] 松塚貴英, “ファイナンスへの計量分析”, 情報処理学会論文誌 Vol.49 No.7 2360-2371, July 2008.
- [4] Erik Pasternak, Rachel Fenichel, Andrew N. Marshall, “Tips for Creating a Block Language with Blockly”, IEEE Blocks and Beyond Workshop, July 2007.