

1. はじめに
2. ビジュアルプログラミング
3. おわりに

ビジュアルプログラミング

情報基盤工学講座 横井 稜

1. はじめに
2. ビジュアルプログラミング
3. おわりに

October 16, 2020

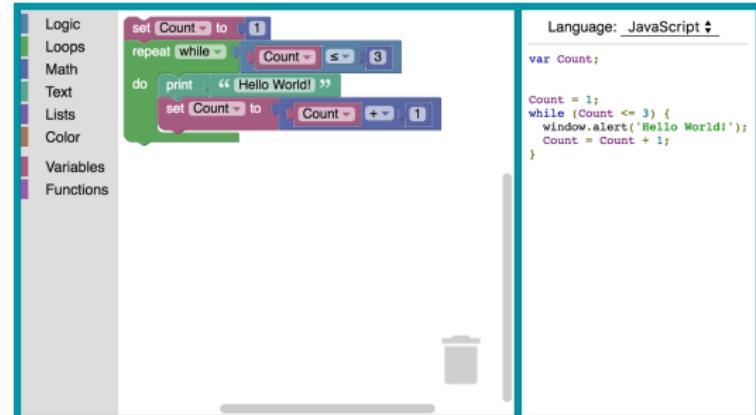
はじめに

2/12

本研究

- はじめに
- ビジュアルプログラミング
- おわりに

近年、企業などでは世間に溢れる様々な情報を収集し、ビッグ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。しかし、そのデータを機械により処理する知識を得るには莫大な時間が必要である。本研究の目的は、データ処理の未経験者でもビッグデータを解析できるようにすることである。



The screenshot shows a Scratch-like programming interface. On the left, there's a sidebar with categories: Logic, Loops, Math, Text, Lists, Color, Variables, and Functions. In the center, a script is being edited:

```
set Count to [1]
repeat (while [Count ≤ 3])
    do [print "Hello World!"]
    set Count to [Count + 1]
```

On the right, the code is displayed in JavaScript:

```
var Count;

Count = 1;
while (Count <= 3) {
    window.alert('Hello World!');
    Count = Count + 1;
}
```

VPL

- はじめに
- ビジュアルプログラミング
- おわりに

プログラムをテキストで記述するのではなく、視覚的なオブジェクトで記述するプログラミング言語のこと。視覚的でわかりやすいものが多いため、プログラムの組み立て方を学ぶのに有効であると注目されている。



メリット

1. はじめに
2. ビジュアルプログラミング
3. おわりに

- ① 直感的に操作できる
- ② 学習が比較的簡単（記憶する必要がほぼない）
- ③ 文字列を経験豊富なプログラマーが認識するように表示してくれる
- ④ 間違いにくい

デメリット

- ① プログラムの修正に時間がかかる
- ② コードを検索しにくい
- ③ テキストとブロックの対訳を作るのが難しい
- ④ 本格的なプログラミング学べない

ブロックタイプ

- ① ブロックを使う
- ② ブロックを並べる順序や論理構造の作り方は、テキスト言語に似ている
- ③ オープンソースのものが多い

フロータイプ

- ① フローチャートを使う
- ② ブロックタイプと見た目の違いがあるだけで基本的なプログラムの作り方は同じ

独自ルールタイプ

- ① 文字や数字で指示や論理構造を作ったり考えたりするのではなく、より直感的な独自の手法でプログラムを作る
- ② テキストプログラミングに移行するのは難しい

システム完成図

6/12

- はじめに
- ビジュアルプログラミング
- おわりに

目標として、MAGELLAN BLOCKS でできる重回帰分析をできるようにする。

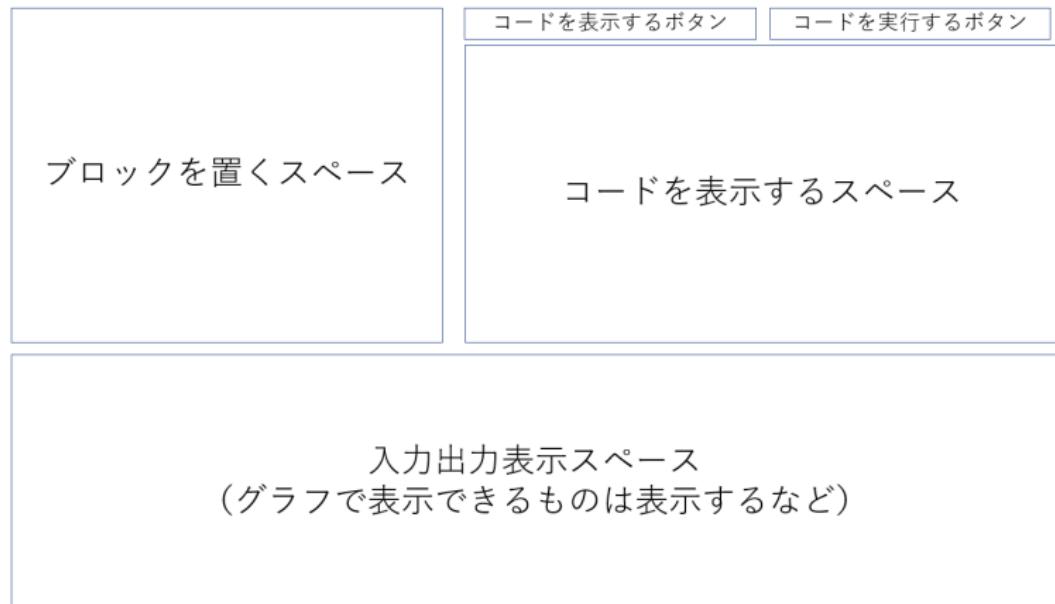


図1 システムの概略図

線形回帰

7/12

1. はじめに
2. ビジュアルプログラミング
3. おわりに

線形回帰

線形回帰とは、教師あり学習の 1 つで 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である。単回帰分析が 1 つの特徴量と 1 つの連続値の目的変数で行われ、重回帰分析が複数の説明変数と 1 つの目的変数で行われる。線形回帰を行うことで、未来の数値を予測することができる。

	E	F	G	H	I	J	K	L
ual : "chlorides	"free sulfu	"total sulf	"density"	"pH"		"sulphate:	"alcohol"	"quality"
1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5
1.2	0.065	15	21	0.9946	3.39	0.47	10	7
2	0.073	9	18	0.9968	3.36	0.57	9.5	7
6.1	0.071	17	102	0.9978	3.35	0.8	10.5	5

作るシステムのフローチャート

8/12

1. はじめに
2. ビジュアルプログラミング
3. おわりに

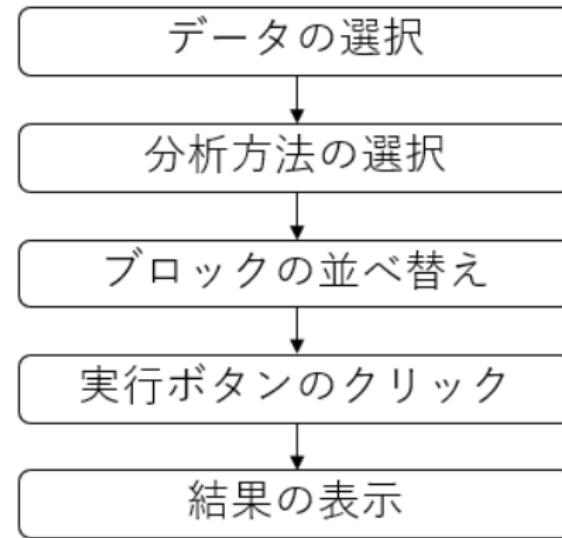


図3 クライアント側のフローチャート

今回研究でやったこと

9/12

- はじめに
- ビジュアルプログラミング
- おわりに



図 4 blockly

python のブロックのデモ

10/12

1. はじめに
2. ビジュアルプログラミング
3. おわりに

1. はじめに
2. ビジュアルプログラミング
3. おわりに

クラス分類・クラスタリング

クラス分類：事前に正解が決まっている属性を適切に付与する問題
(例：手書き文字認識)

クラスタリング：いくつかのかたまりにグループ分けする問題、絶対的な正解はない (例：顧客のグルーピング)

アソシエーション分析

アソシエーション分析は、店頭で購入するカゴの中身を推測するような分析方法です。基本的なアソシエーション分析では、データベース内の莫大なデータのなかから、パターンや関連性を引き出すマシーンラーニングモデルを用いて行われます。

まとめ

1. はじめに
2. ビジュアルプログラミング
3. おわりに

- ① python を用いた回帰分析ブロックの作成**
- ② 追加するデータ処理ブロックの選定**

今後の課題

- ① カスタムブロック及びシステムの作成**
- ② サーバーの管理及び運営**