

オンライン機械学習フレームワーク Jubatus による ライフログからの情報抽出

黒崎 裕子[†] 山下 暁香[†] 小口 正人[†]

[†] お茶の水女子大学 〒112-8610 東京都文京区大塚 2-1-1

E-mail: [†]{yuko-k,akika}@og1.is.ocha.ac.jp, ^{††}oguchi@computer.org

あらまし 近年ではカメラやセンサ等の利用により、一般家庭でもライフログの取得が容易になった。防犯対策やセキュリティ、お年寄りや子供のための安全サービスを目的として、ライフログ解析アプリケーションは多く開発されてきたが、実際に一般家庭で採用する場合、サーバやストレージを設置して解析までを行うことは難しい。また、ライフログは常時蓄積され、ビッグデータに発展するため、クラウド上での処理が好ましいと考えられる。本研究ではクラウド上の機械学習、ストレージに注目し、クラウド上でのライフログの解析を試みた。今回の実験ではクラウド上での機械学習に焦点をあて、ライフログは常時生成されるものであることから、オンライン機械学習フレームワーク Jubatus を用いてライフログ解析アプリケーションの実装を行う。Jubatus に入力する学習用動画データの種類によってどのように学習精度が変化するかを評価し、また動画データのみによる機械学習だけでなく、ベイズを用いた解析方法でもライフログの解析を行い、学習精度等の比較を行った。

キーワード オンライン機械学習, 情報抽出

Information Extraction from the Life Log with an On-Line Machine Learning Framework Jubatus

Yuko KUROSAKI[†], Akika YAMASHITA[†], and Masato OGUCHI[†]

[†] Ochanomizu University Otsuka 2-1-1, Bunkyo-Ku, Tokyo 112-8610 Japan

E-mail: [†]{yuko-k,akika}@og1.is.ocha.ac.jp, ^{††}oguchi@computer.org

1. はじめに

近年ではカメラやセンサ等の利用により、一般家庭でもライフログの取得が容易になった。防犯対策やセキュリティ、お年寄りや子供のための安全サービスを目的として、ライフログ解析アプリケーションは多く開発されてきたが、実際に一般家庭で採用する場合、サーバやストレージを設置して解析までを行うことは難しい。文献 [1] の研究ではスタンドアロン環境でのライフログ解析アプリケーションが開発され、実験が行われた。しかし、ライフログは常時蓄積され、ビッグデータに発展するため、クラウド上での処理が好ましいと考えられる。

本研究ではクラウド上の機械学習、ストレージに注目し、クラウド上でのライフログの解析を試みた。今回の実験ではクラウド上での機械学習に焦点をあて、ライフログは常時生成されるものであることから、オンライン機械学習フレームワーク Jubatus [2] を用いてライフログ解析アプリケーションの実装を行う。図 1 に示すように、家庭でライフログを取得しながら、ク

ラウド上でライフログの蓄積と解析を行い、解析結果を返す、というシステム環境作りを目的としている。

以下、2 節では機械学習について述べ、3 節では Jubatus への予測用データをベイズ分類器の結果ノードのビット、画素データとした場合、それぞれの実験について述べる。4 節では実験環境について説明し、5 節で実験結果を紹介する。最後に、6 節でまとめを述べる。

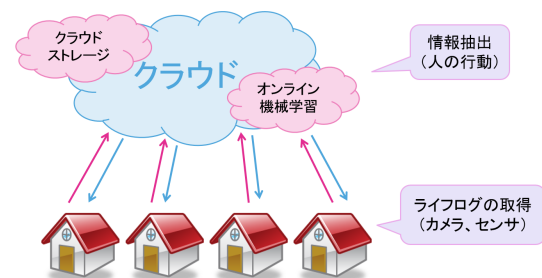


図 1 研究目的の概要図

2. 機械学習

機械学習とはデータ解析から有用な規則を抽出し、アルゴリズム化することである。機械学習には大きく分けて2つあり、教師有り学習と教師なし学習である。入力 x から出力 y への関数を x_i, y_i をもとに行う機械学習を教師あり学習といい、入力 x_i のみから学習するものを教師なし学習という。

また、学習用データを貯めてまとめて学習することをバッチ処理という。多くの機械学習ではバッチ処理が採用されている。これを対比して、データを1つ受け取るたびに学習を行うものをオンライン機械学習という。

2.1 Mahout

機械学習のアルゴリズムを一から実装することは難しく、多くの機械学習のアルゴリズムを実装したライブラリが開発され、公開されている [3]。その1つが Mahout である。Mahout の大きな特徴は、Apache Hadoop を内部で利用しているところである。Mahout ではスケーラビリティを最優先に考えられており、Hadoop には分散処理機能が含まれているため、Mahout を利用すれば、分散処理が可能になる。

2.2 Jubatus

Jubatus とは、文献 [2] で紹介されているように、NTT SIC と Preferred Infrastructure により共同開発された、オンライン機械学習フレームワークである。本来トレードオフの関係であった「ストリーム (オンライン) 処理」「並列分散処理」「深い解析」の要素を満たすオンライン機械学習フレームワークである。

オンライン機械学習をそのまま分散処理すると同期コストが大きくなるという問題が生じるが、Jubatus ではデータ自体は共有せず、学習後のモデルのみを緩やかに共有することで、並列分散処理を可能にしている。また、多値分類、線形回帰、推薦 (近傍探索)、グラフマイニング、異常検知、クラスタリングなど多くの解析手法をサポートし、それに加え、画像や HTML、Twitter のような非定型データも扱うことが可能で、より深い解析を行うことができる。

2.3 Hadoop/Mahout と Jubatus の比較

Jubatus と Hadoop/Mahout の共通点はスケーラブルであり、コモディティサーバ上で動作することである [2]。バッチ処理を得意とする Hadoop は、機械学習が MapReduce パラダイムにあまり適合していないため、洗練された機械学習機能を備えていない。Mahout は Hadoop-based な機械学習ツールであるが、オンライン処理は対象外である。Hadoop/Mahout と比較して、Jubatus はスケーラビリティに加えて、オンライン処理も兼ね備えている。ライフログ解析システムを構築するためには、オンライン処理、スケーラビリティ、深い解析が必要となるため本研究では Jubatus を用いて実験を進める。

3. 実験概要

Jubatus の基本的な操作として、学習用データを準備し、Jubatus に先に学習させる操作を UPDATE という。それぞれのモデルの共有、データ更新を MIX といい、予測データを Jubatus

に渡して予測結果を受け取る操作を ANALYZE という [2]。

本研究では Jubatus に渡す学習用データを動画画像の画像データ、ベイズ分類器の結果ノードのビットの2パターン用意する。それぞれの学習用データを読み込むようなライフログ解析アプリケーションの実装を行う。

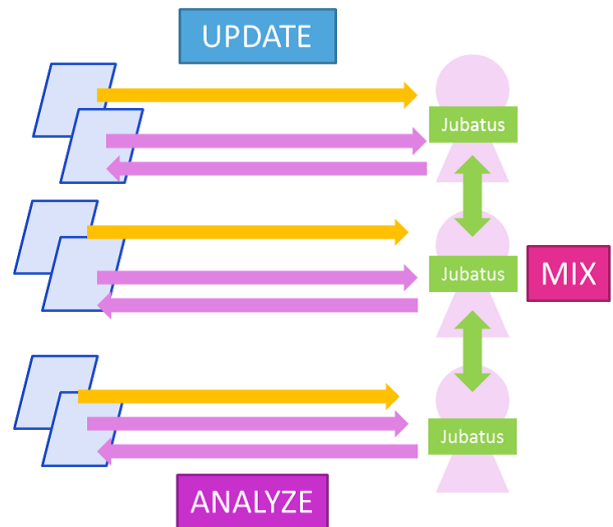


図2 Jubatus の学習の流れ

3.1 学習用データ：ベイズ分類器の結果ノードのビット

文献 [1] で紹介されたライフログ解析アプリケーションのプログラムに Jubatus を組み込んでライフログ解析アプリケーションの実装を行った。

まず、ベイズ分類器について説明する。ベイズ分類器とは、与えられた CPT (確率分布表) をもとに結果から原因を予測するモデルである。このライフログ解析アプリケーションでは原因を「人の行動」、結果を「カメラやセンサの反応」とする。

A_i を人の行動 ($i=2$ で2つの動作「ドアを開ける」「イスにすわる」を用意)、3つのビット情報を ($R1, R2, R3$)=(カメラ1, カメラ2, センサ) とする。図3のように、2台の異なる角度に設置されたカメラの動画画像とセンサデータを取得する。動画画像データには画像処理を施して、連続する2枚のキャプチャの差分の重心を求め、その重心が定義した物体に何回重なったかを数える。その回数が予め定義した閾値を超えた場合、2台のカメラのビットを立てる。センサも同様に、値の変化量が閾値より大きい場合ビットを立てる。

予め与えられた CPT (図4) を用いて、公式

$$P(A_i|R1, R2, R3) = \frac{P(A_i)P(R1, R2, R3|A_i)}{P(R1, R2, R3)}$$

の計算を行い、最大にする A_i をもっともらしい原因とし、結果ノードのビット ($R1, R2, R3$) と A_i の組み合わせを学習用データとして Jubatus に学習させる (図5)。ただし、データ処理の計算を行う際は、以下のように変形して計算を行う。

$$\underset{i}{\operatorname{argmax}} P(A_i|R1, R2, R3)$$

$$\begin{aligned}
&= \underset{i}{\operatorname{argmax}} \frac{P(A_i)P(R1, R2, R3|A_i)}{P(R1, R2, R3)} \\
&= \underset{i}{\operatorname{argmax}} P(A_i)P(R1, R2, R3|A_i) \\
&= \underset{i}{\operatorname{argmax}} P(R1, R2, R3|A_i)
\end{aligned}$$

入力したカメラの動画データやセンサデータから画像処理やセンサ処理を通してビット情報を取得し、学習を終えた Jubatus に予測用データとして渡すことで、人がどのような行動をしたのかという重みを出力する。

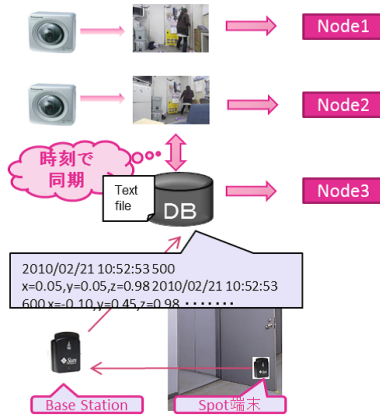


図 3 実験概要 1(ベイズ分類器の結果ノードビットの学習)

R \ A	A ₁	A ₂	A ₃	R \ A	A ₁	A ₂	A ₃	R \ A	A ₁	A ₂	A ₃
0	0.1	0.4	0.5	0	0.1	0.9	0.8	0	0.01	0.9	0.9
1	0.9	0.6	0.5	1	0.9	0.1	0.2	1	0.99	0.1	0.1

R \ A	A ₁	A ₂	A ₃	R \ A	A ₁	A ₂	A ₃	R \ A	A ₁	A ₂	A ₃
0	0.9	0.1	0.9	0	0.9	0.9	0.9	0	1.0	1.0	1.0
1	0.1	0.4	0.1	1	0.1	0.1	0.1	1	0.0	0.0	0.0

R \ A	A ₁	A ₂	A ₃	R \ A	A ₁	A ₂	A ₃	R \ A	A ₁	A ₂	A ₃
0	0.9	0.9	0.1	0	0.5	0.9	0.6	0	0.9	0.9	0.01
1	0.1	0.1	0.9	1	0.5	0.1	0.4	1	0.1	0.1	0.99

図 4 与えられた CPT

```

train_data.push_back(make_pair("ドアを開ける",make_datum(0,1,1)))
train_data.push_back(make_pair("ドアを開ける",make_datum(1,0,1)))
.
.
train_data.push_back(make_pair("イスに座る",make_datum(0,1,0)))
train_data.push_back(make_pair("イスに座る",make_datum(1,0,1)))
train_data.push_back(make_pair("イスに座る",make_datum(1,1,1)))
.
.

```

図 5 ベイズ分類器の結果ノードビットの学習

3.2 学習用データ：画像データ

学習用データを動画のキャプチャ画像データとする場合、人の行動を写した画像データを Jubatus に学習させる。Jubatus 0.5.1 は特徴ベクトル変換のバイナリ入力に対応しており、画像のインプットは可能である。しかし、分類などのコードは一般公開されておらず、プラグインを開発し用いなければならない。クライアント側で OpenCV を用いた特徴抽出を行うことも考えられるが、研究目的としてクライアント側の処理は軽くしたいので、Jubatus 側での処理を考えたい (図 6)。今回の実験では、まずプラグインの開発実装を行い、簡単なサンプルプログラムを組んでプラグイン動作確認を行う。

実装したプラグインでは OpenCV を用いて特徴量を抽出している [4]。また、JSON ファイルにおいてプラグインを使用する設定を行う。

1 つ目の実験と学習精度比較を行うため、2 つの行動「ドアを開ける」「イスにすわる」を写した画像を用意し、図 7 のように画像にラベル付けを行って、Jubatus に学習させる。予測用データには画像データを用い、Jubatus に画像データを直接渡し、人がどのような行動をしたのかという重みと予測結果を出力する。

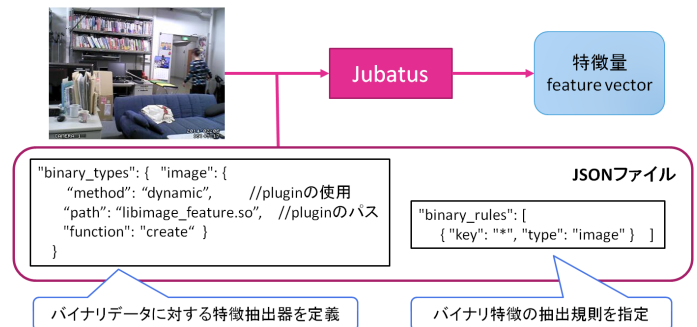


図 6 実験概要 2(画像データの学習)



図 7 画像データの学習

4. 実験環境

マシンには Intel Core 2 Duo E6400 2.13GHz を使用し、OS には Jubatus を公式でサポートしてる Ubuntu 12.04 LTS を用いた。使用ツールとしては、オンライン機械学習フレームワーク Jubatus 0.5.1、画像処理を施す際、OpenCV 2.4.7 を使用した。

また動画取得には WEB カメラの UCAM-DLV300T を使用した [5]。画像サイズは 640 × 480 とする。

5. 実行結果

5.1 学習用データ：ベイズ分類器の結果ノードのビット

既存研究のアプローチを Jubatus 実装したプログラムを実行した。「イスに座る」という動作を含んだ動画データとその時のセンサデータを読み込ませ、動画の 1 コマ目から物体を定義した (図 8)。CPT を用いベイズ分類器で計算すると、ベイズ分類器の結果として図 9 のように結果ノードのビット (1,1,0) が学習用データとして Jubatus に入力された。その結果として、それぞれの動作の重みが出力され、「イスに座る」が 1 番重みが大きいという結果が得られた。

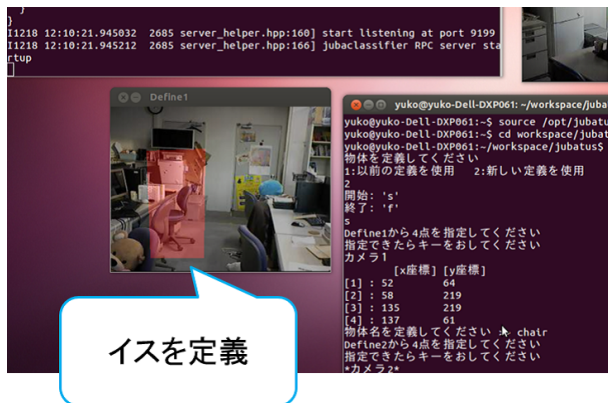


図 8 物体の定義

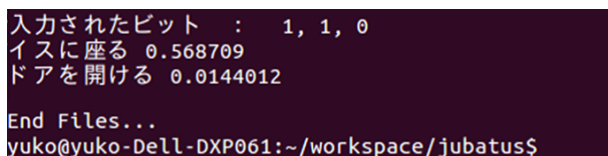


図 9 実行結果 1(ベイズ分類器の結果ノードビットの学習)

5.2 学習用データ：画像データ

学習用データに使用した画像を同じ角度のカメラから撮影した「イスに座る」という動作を写した画像を Jubatus に入力すると、Jubatus 内でプラグインを用いた特徴抽出が行われ、図 10 のようにそれぞれの動作の重みが出力された。「イスに座る」が 1 番重みが大きいという結果が得られた。

これにより、実装したパイナリデータの特徴抽出プラグインの動作も確認できた。

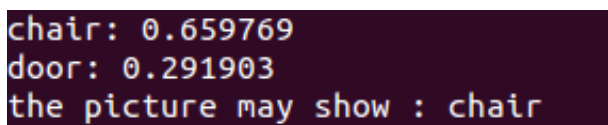


図 10 実行結果 2(画像データの学習)

6. まとめと今後の課題

取得したライフログを Jubatus にのせて実際に動かした。学習用データをベイズ分類器の結果ノードビットとした場合では、正しい結果は得られたが、予想より重みの差がつかなかったため、閾値などプログラムを書きかえて、より精度を高くすることを目指すつもりである。また、学習用データを画像データとした場合では、Jubatus のプラグインを開発し、Jubatus 内での特徴抽出を行って、正しい結果が得ることができた。現段階では、静止画像データのみでの解析までしか行うことができないため、動画像データを入力しても解析できるような実装を行いたい。

今後の実験では、用意する画像データの質の違いによる学習結果の比較、学習データが画像データ、結果ノードのビット、それぞれの場合で学習精度の比較を行いたい。さらに、現段階ではサンプル動画や画像を用いて実験を行っているため、本格的なオンライン機械学習ができていない。今後は取得したライフログをどのようにクラウドストレージへ移行するか検討し、オンライン、クラウド化に向けた、クラウド上における処理方法の検討も行ってきたい。

文 献

- [1] 山下暁香, 小口正人: 無線通信の状態に基づくデータ品質変化時のライフログ解析アプリケーションの動作評価, 情報処理学会論文誌, コンシューマ・デバイス&システム, vol.3, No.1, pp.87-97, 2013 年 3 月。
- [2] オンライン機械学習向け分散処理フレームワーク Jubatus <http://jubat.us/ja/>
- [3] 機械学習ライブラリ Mahout <http://mahout.apache.org/>
- [4] OpenCV <http://opencv.org/>
- [5] UCAM-DLV300T <http://www2.elecom.co.jp/multimedia/pc-camera/ucam-dlv300t/>