

オンライン機械学習フレームワーク Jubatus を用いた ライフログ解析に関する一検討

黒崎 裕子[†] 山下 暁香[†] 小口 正人[†]

[†]お茶の水女子大学

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。既存研究 [1] 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。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。図 1 に表わすように、家庭でライフログを取得した後、クラウド上でライフログ解析、保存を行い、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。

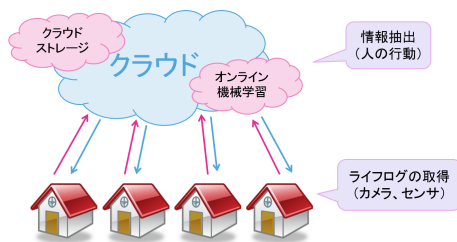


図 1: 研究目的の概要図

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、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、教師なし学習がある。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 Jubatus

Jubatus とは、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 [2]。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。

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 (UPDATE)。その学習データを更新し (MIX)、予測データを準備し、Jubatus に渡して予測結果を受け取る (ANALYZE)。本研究では Jubatus に渡す学習用データを動画像の画素データ、ベイズ分類器の結果ノードのビットの 2 パターン用意し、それぞれの学習精度の評価、比較目指している。

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 ($i=3$ で 3 つの動作「ドアを開ける」「イスにすわる」「机をふく」を用意)、3 つのビット情報を (R_1, R_2, R_3) = (カメラ 1, カメラ 2, センサ) とする。図 2 のように、2 台の異なる角度に設置されたカメラの動画像とセンサデータを取得する。動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 を用いて、公式 $P(R1,R2,R3 | A_i)$ の計算を行い、計算された $P(R1,R2,R3 | A_i)$ が最大になる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 3 参照)

予測用データとして、カメラやセンサから取得したビット情報を、学習を終えた Jubatus に渡すことで、人がどのような行動をしたのかという重みを出力する。

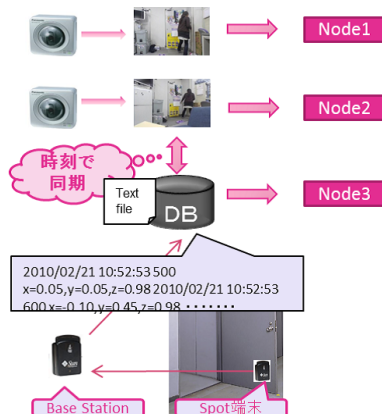


図 2: 実験概要

```
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(0,0,1)))
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(1,1,0)))
.
.
.
train_data.push_back(make_pair("机をふく",make_datum(1,0,0)))
.
.
.
```

図 3: 学習用データ：ベイズ分類器の結果ノードのビット

3.2 学習用データ：画素データ

学習用データを動画の画素データとする場合、人の行動を写した画像の画素データを Jubatus に学習させる。その後、予測用データとして動画を Jubatus に渡し、人がどのような行動をしたのかという重みを出力する。

1 つ目の実験と学習精度比較を行うため、3 つの行動「ドアを開ける」「イスにすわる」「机をふく」を写した画像を 10 枚ずつ用意し、Jubatus に学習させる。予測用データには動画画像からキャプチャを取り出して、その画素データを Jubatus に渡す。

なお、この実験については現在実装段階である。

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 12.04LTS 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 0.45、画像処理を施す際、OpenCV 2.4.7 を使用した。

5. 実行結果

「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て、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 4 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た。「イスに座る」が 1 番重みが大いという結果が得られた。

```
入力されたビット : 1, 1, 0
机を拭く 0.20609
イスに座る 0.669899
ドアを開ける 0.225824

End Files...
yuko@yuko-Dell-DXP061:~/workspace/jubatus$
```

図 4: 実行結果

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。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、画素データのみを入力して結果を出力させるプログラムの実装を行い、用意する画素データの質の違いによる学習結果の比較、学習データが画素データ、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を用いて実験を行っているので、本格的なオンライン機械学習ができておらず、今後はオンライン、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。

参考文献

- [1] 山下暁香, 小口正人: 無線通信の状況に基づく入力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, DICOMO2012, 6H-5, pp.1731-1737, 2012 年 7 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/>