

クラウド上でのライフログ解析のための オンラインフレームワーク Jubatus を用いた アプリケーション実装

黒崎 裕子¹ 竹房 あつ子² 中田 秀基² 小口 正人¹

概要：

近年ではカメラやセンサ等を手軽に利用できるようになり、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が数多く開発されてきたが、実際に一般家庭で採用する場合は、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ため、クラウドでのライフログ解析が必要となる。しかしながら、動画のようなセンサーデータはデータ量が大きいので、特徴量抽出等の前処理は各センサー側で行い、特徴量のみをクラウド側で収集して解析する手法も考えられる。本研究では、ライフログ解析の前処理をセンサー側で行う場合とクラウド側で行う場合での性能比較を行うことを目的として、評価対象となるオンライン学習アプリケーションを実装した。実装したアプリケーションでは、カメラが動体検知すると前処理として撮影した画像の特徴量抽出を OpenCV を用いて行い、その特徴量をもとに Bag-of-Features 手法を用いてベクトル化したデータに変換する。そのデータを入力値として、オンライン機械学習フレームワーク Jubatus を用いて学習し、人の行動の判別を行う。実験から、カメラ画像を前処理したデータを入力値として、Jubatus を用いて行動判別が行えることが確認され、実装したアプリケーションの動作確認ができた。

Implementation of a Lifelog Analysis Application running on Clouds using Online Machine Learning Framework

YUKO KUROSAKI¹ ATSUKO TAKEFUSA² HIDEMOTO NAKADA² MASATO OGUCHI¹

1. はじめに

一般家庭でもカメラやセンサ等を手軽に利用できるようになり、ライフログの取得が容易になったことから、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が数多く開発されてきた。しかし、実際に一般家庭で採用する場合は、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、ライフログは常時生成され、そのデータ量は膨大になるため、クラウド上での解析が好ましいと考えられる。しかしながら、動画のようなセンサーデータは

データ量が大きく、全てのデータをクラウド側に送るのは非効率であると考えられる。よって、ライフログ解析アプリケーションを設計する場合、センサーデータをすべてクラウド側に送って処理する方法と、前処理はセンサー側で行い、前処理済みの少量データをクラウド側に送って処理する方法が考えられる。

本研究では、ライフログ解析の前処理をセンサー側で行う場合とクラウド側で行う場合での性能比較を行うことを目的として、リアルタイムに解析を行うライフログ解析アプリケーションを実装した。ライフログは常時生成されるものであることより、多くの機械学習ツールの中からオンライン機械学習フレームワーク Jubatus を用いてアプリケーションの実装を行った。Jubatus では学習モデルのみの共有によって、並列分散処理によるオンライン学習が可

¹ お茶の水女子大学
Ochanomizu University, Bunkyo, Tokyo 112-8610, Japan

² 産業技術総合研究所
AIST, Tsukuba, Ibaraki 305-8568, Japan

能である。

実装するライフログ解析アプリケーションは、カメラが動体検知すると前処理として画像から特徴抽出を行い、その結果を入力として Jubatus で解析するように設計した。我々の既発表研究 [5] では、前処理部分から解析まですべて Jubatus が行っていたため、データ取得から解析結果の出力に時間がかかっていた。本研究で実装したアプリケーションでは、前処理として撮影した画像を画像処理ライブラリ OpenCV の SURF を用いて特徴点抽出を行い、そのデータをもとに Bag-of-Features 手法を用いてベクトルデータにする。そのベクトルデータを入力値として Jubatus に学習させ、人の行動の判別する。実験から、カメラ画像を前処理したデータを入力値とし、Jubatus を用いて行動判別が行えることが確認された。これにより実装したアプリケーションの動作確認ができた。

2. オンライン機械学習 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。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 [1]。

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。

Jubatus と分散処理基盤 Apache Hadoop、機械学習フレームワーク Mahout の共通点はスケラブルであり、コモディティサーバ上で動作することである [1]。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 上で動作する機械学習フレームワークであるが、オンライン処理は対象外である。一方、Jubatus はスケラビリティに加えて、オンライン処理が可能である。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、スケラビリティ、深い解析が必要となるため本研究では機械学習フレームワークとして Jubatus を用いる。

Jubatus の基本操作について説明する。図 1 に示すように、Jubatus には大きく 3 つの操作がある。まず Update 操作で各マシンが学習、解析を行い差分データが蓄積される。各マシンの差分データを Mix 操作で混ぜ合わせることでモデルデータを作成し、全てのマシンで共有され、Analyze 操作でこれらのモデルデータを利用して分析を各

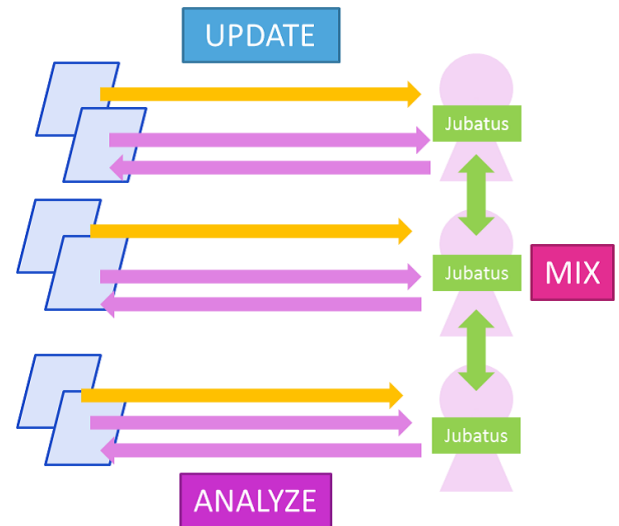


図 1 Jubatus の基本操作

マシンで行われる。Mix については Update 回数や時間などの条件が満たされた時に自動的に呼び出されるため、クライアントは Update 操作と Analyze 操作を呼び出すだけで Mix 操作が可能である。

Jubatus は解析の時に datum という key-value データ形式を用いる。datum には 3 つのタイプの key-value が存在する。value が文字列の文字列データ、value が数値の数値データ、最後に value が任意のバイナリデータがあり、key は 3 タイプとも文字列である。バイナリデータには画像や音声などのマルチメディアデータなど、任意のバイナリデータを入れることが可能である。この 3 つのデータから、機械学習を行う際に必要となる特徴量を Jubatus のデータ変換モジュールが抽出する。

また、Jubatus の特徴ベクトル変換器は、この特徴抽出処理を JSON 形式ファイルでカスタマイズすることが可能であり、特徴抽出器にはプラグインを利用することができる。プラグインは動的ライブラリファイル(.so ファイル)からなり、JSON 形式ファイルでパスを通すことによって利用可能である。本研究でも、プラグインを用いて特徴ベクトル変換器をカスタマイズし、実験を行う。

3. Jubatus と OpenCV を用いたライフログ解析アプリケーションの設計

本節では本研究で実装したライフログ解析アプリケーションの概要を説明する。実装したアプリケーションを大きくわけて次の 3 つのプロセスに分かれている。

- (1) Motion を用いたカメラからの画像取得
- (2) Bag-of-Features を用いた特徴抽出
- (3) Jubatus での解析

以下に各プロセスの詳細について述べる。

3.1 Motion を用いたカメラからの画像取得

カメラからの動体検知と画像取得には、Web カメラサーバソフト Motion[3] を使用した。Motion で動体検知すると取得した画像を随時保存し、画像を保存した際に指定したスクリプトファイルを実行するように設定した。スクリプトファイルでは、取得した画像ファイル名を (2)Baf-of-Features を用いた特徴抽出プロセスに転送する。

3.2 Bag-of-Features を用いた特徴抽出

Jubatus での解析の前処理として、OpenCV[2] を用いた特徴量の抽出と Bag-of-Features[4] による画像データのベクトル化を行い、ベクトル化したデータを (3)Jubatus での解析プロセスに転送する。Bag-of-Features とは、画像から得られた局所特徴量の集合から、あらかじめ複数画像の特徴量データから K-means 手法を用いて特徴量をクラスタリングした辞書データをもとに、各グループに属する特徴量をもつ特徴点の数をヒストグラム化する手法である。Bag-of-Features の抽出手順は以下のようになる。

まず、OpenCV を用いて画像から局所特徴量を抽出する。局所特徴量にはいくつか種類がある。その中でも有名な SIFT は、照明変化や回転、拡大縮小に不変な頑強な特徴量である。その SIFT より認識精度は少し落ちるが、特徴点検出の処理を軽量化、高速化したものが SURF である。本研究では、局所特徴量に SURF 特徴量を用いた。SURF では keypoint と呼ばれる画像中の特徴的な点をいくつか抽出する。各 keypoint は 128 次元の特徴ベクトルとなるが、抽出される keypoint の数は画像によって異なるため、画像全体の特徴ベクトルとして機械学習でそのまま使用するのは困難である。局所特徴量をクラスタリングして各クラスターの中心ベクトルを Visual Word と呼ばれる特徴的なパターンとし、辞書を作成する。この辞書を使ってある画像から抽出された特徴ベクトル群を Visual Word にマッチさせ、画像全体の特徴を特徴パターンの頻度をヒストグラムで表現する。このヒストグラムを Bag-of-Features と呼ぶ。一度辞書を作成してしまうと、Visual Word 数は一定であるため、各画像の特徴量を同じサイズのベクトルデータに変換することができる。

次に、生成したベクトルデータを Jubatus で扱う形式に変換する。2 節で説明したように、Jubatus は解析時に datum というデータ形式を使用する。よって、作成したヒストグラムを datum に変換する。ヒストグラムの各値を 1 つずつリスト形式で datum に格納し、Jubatus へ転送可能なデータに変換する。最後に、Jubatus サーバとコネクションを確立し、画像から取得した特徴ベクトルの格納された datum を Jubatus サーバへ転送する。

3.3 Jubatus での解析

Jubatus では分類、推薦、線形回帰などいろいろな解析

が可能であり、今回は Classifier API を用いて学習と分類を行う。ライフログ解析を行う前に trainAPI を利用し、予め教師あり学習を行う。その学習結果を用いて分類するには、classifyAPI を利用する。Jubatus サーバに送られてきた datum はフィルター、特徴抽出という 2 段階のデータ変換を経て解析される。フィルター処理では、学習に不要なものを取り除き、特徴抽出では、フィルターされたデータから特徴を抽出する。フィルター処理でデータからどのような要素を取り除くか、特徴抽出でどのアルゴリズムを使用し、どのように重み付けをするかは、サーバ起動時に指定する JSON ファイルで設定することが可能である。今回はフィルターはデフォルトを利用し、特徴抽出では与えられた数値をそのまま重みに利用する。分類に使用するアルゴリズムには Adaptive Regularization of Weight vectors を選択した。学習に対する感度パラメータは 1.0 に設定する。Jubatus サーバで 2 段階のデータ変換を終えた後、解析され、解析結果を送信する。

3.4 特徴抽出処理のセンサー側実装とクラウド側実装

ライフログ解析アプリケーションの設計では、特徴抽出処理をセンサー側で行う方針とクラウド側で行う方針が考えられる。ここでは、センサー側実装とクラウド側実装の違いについて説明する。

センサー側の実装は、図 2 に示すようにセンサー側で (1)Motion を用いたカメラからの画像取得プロセスを経て、取得した画像を (2)Baf-of-Features を用いた特徴抽出プロセスに転送し、ベクトル化したデータをクラウド側で実行されている Jubatus サーバへ転送する。Jubatus サーバがデータを受け取ると、クラウド側で (3)Jubatus での解析プロセスで解析が行われる。

また、クラウド側の実装は以下のようになる。図 3 に示すように、センサー側で (1)Motion を用いたカメラからの画像取得プロセスを経て画像を取得し、取得した画像に加工を施さず、そのままクラウドに転送する。クラウド側で画像データを受け取ると、(2)Baf-of-Features を用いた特徴抽出プロセスで画像データのベクトル化を行い、同様にクラウド側で実行されている Jubatus サーバに接続し、(3)Jubatus での解析プロセスで解析が行われる。

4. 実験

本研究は初期実験として今回実装したアプリケーションの動作確認を行う。実験では、実装したアプリケーションを用いて「ドアを開けた」「イスに座った」の 2 種類の人の行動を判別する。本研究では、3 節で述べた (1)、(2)、(3) のプロセスをすべて同一の計算機上で実行した。

マシンには Intel Core 2 Duo E6400 2.13GHz を使用し、OS には Ubuntu 12.04LTS を用いた。また、Jubatus v. 0.5.1, OpenCV v. 2.4.7 を使用した。動画取得には WEB

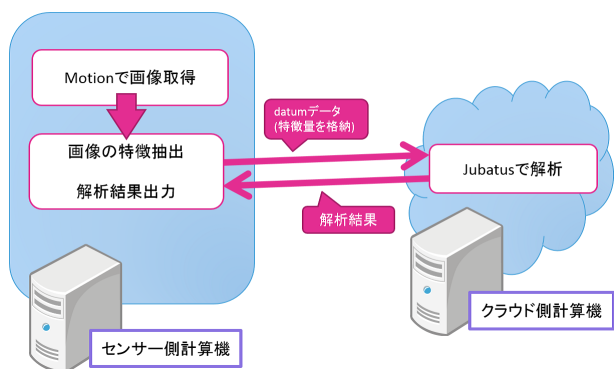


図 2 実装概要 (センサー側での特徴抽出)

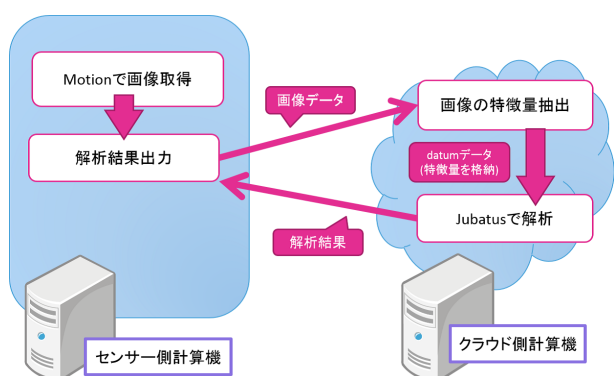


図 3 実装概要 (クラウド側での特徴抽出)

カメラの UCAM-DLV300T を使用し、画像サイズは 640 × 480 とした。特徴抽出における Visual Words 数は 100 に設定し、辞書作りを行った。

実行結果は図 4 のようになった。実験では、予め 100 枚の画像を用いて「ドアを開けた」状態と「イスに座った」状態を学習させた後、カメラから得られた図 4 上のような画像データから特徴抽出処理を行った。図 4 下の端末画面に Bag-of-Features で生成したベクトル値が表示されている。このベクトル値を datum に格納して Jubatus に転送した。「イスに座る」という画像をカメラから取得し、前処理を行って Jubatus に転送して解析した結果、「chair」という正しい分類結果が出力された。

5. まとめと今後の課題

本研究ではライフログ解析の前処理をセンサー側で行う場合とクラウド側で行う場合での性能比較を行うことを目的として、オンライン解析可能なライフログ解析アプリケーションの実装を行った。実装したアプリケーションでは、画像データに対して Bag-of-Features 手法を用いて前処理を行い、前処理を行ったデータを入力値として Jubatus サーバで学習することで、人の行動判別が確認できた。今後は、前処理のセンサー側処理とクラウド側処理の性能比較を行い、転送量と学習速度のバランスについて検証する。

そのために、発生した大量のライフログをどのようにストレージに保存していくかの検討を行いたい。現在、カメ



図 4 実装結果 (センサー側での特徴抽出)

ラより取得した画像データはローカル PC 上に蓄積される。ライフログは常時生成されること、扱うデータがデータ量の大きい画像データであることを考慮して、クラウドストレージの利用を考える。どのストレージを使用するかを含め、取得したライフログの保存手法について検討したい。また、センサー側からライフログ参照、検索機能を追加するなど、アプリケーションの機能拡張も行っていく。さらに、学習精度についても評価実験を行い、画像の品質や動体検知の精度を上げた実装も行う。

6. 謝辞

本研究を進めるにあたって、Preferred Infrastructure の海野裕也氏、河原 一哉氏、阿部 厳氏より大変有用なアドバイスをいただきました。深く感謝いたします。

参考文献

- [1] 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/>.
- [2] OpenCV : <http://opencv.org/>.
- [3] Motion : <http://www.lavrsen.dk/foswiki/bin/view/Motion/>.
- [4] T. Nagahashi, H. Fujiyoshi, "Object Category Recognition by Bag-of-Features using Co-occurrence Representation by Foreground and Background Information", Machine Vision Applications, pp.413, 2011.
- [5] 黒崎裕子, 山下暁香, 小口正人: "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", DEIM 2014, D3-4, 2014 年 3 月.