ROS 環境上における機械学習タスク実行モジュールの実装と評価

An Implementation and Evaluation Machine Learning Modules on ROS Environment

阿部 秀尚 [1]　森田 武史 [2]　山口 高平 [2]

Hidenao Abe[1], Takeshi Morita[2], and Takahira Yamaguchi[2]

[1] 文教大学情報学部情報システム学科

[1]Department of Information Systems, Faculty of Information and Communications, Bunkyo University

[2] 慶應義塾大学理工学部管理工学科

[2] Department of Administration Engineering, Faculty of Science and Technology, Keio University

Abstract: In order to achieve more interactive operations for human-robot services, it is necessary to handle each situation based on various sensor data. Many machine learning algorithms have been developed to build learning modules for predicting proper decisions more accurately from stored datasets. In this paper, we designed and implemented the modules for ROS (Robot Operation System) to perform machine learning algorithms. As for designing the machine learning process, we separate each machine learning algorithm into the two phases: training phase for learning models, and prediction phase for a test dataset by using any learning model. Then, the modules have been implemented the ROS modules. By using these modules, we evaluated the performances on the ROS environment. The processing costs and accuracies of the machine learning algorithms such as k-nearest neighbors, decision trees learner, support vector machine learner, naïve bayes, and Random Forest are compared to those of convolutional neural networks for recognizing the categories of the images from the Caltech-101 image set.

1 はじめに

機械学習アルゴリズムを利用した AI サービスの認知度が広まるにつれ，その適用可能な応用に期待が高まっている．しかしながら，実際の応用事例に機械学習アルゴリズムによる処理を組み込むことは容易ではなく，対象業務プロセスに基づき適切なタスクとして機械学習タスクを適用することが必要である．これに対し，我々は，対人ロボットサービスを実現する ROS (Robot Operating System)[1]上のモジュールとして，業務プロセス記述レベルからロボットの動作レベルまでのプロセスを階層的に記述できる環境 PRINTEPS[2]の開発を行ってきた．ロボットサービス記述において，業務プロセスを明記する知識体系化は先行研究[3]においても行われてきたが，機械学習タスクとの統合までには至っていない．

本研究では，ロボットサービスの記述と実行において，より円滑な対人サービスを実現するため，機械学習タスクを業務プロセス記述可能な単位に分割し，ROS モジュールとして実装を行う．

このため，本稿では，ROS 環境上での機械学習タスク実行モジュールの設計と実装について述べ，実装した ROS モジュールによる代表的な機械学習アルゴリズムの画像認識タスクにおける精度および実行時間について示す．さらに，深層学習の一種である畳み込みニューラルネットワークによる同一タスクの実行結果との比較を示す．

以上より，従来から分類学習タスクを行うための学習モデルを構築するために開発されてきた各種機械学習アルゴリズムと深層学習による畳み込みニューラルネットワーク構築の各特徴について考察する．

2 ROS 環境での機械学習タスクの実装

本節では，本研究における ROS 環境に適合した機械学習タスクの設計と実装[4]について具体的に述べる．

ROS は，ロボットの動作実行をプログラミングし，

実行するためのライブラリを備えたミドルウェアである．各実行プログラムは“ノード”と呼ばれ，ノードには常時繰り返し動作を行う“トピック”，非同期の動作を行う“サービス”がある．また，ROS 環境においては，ノード間の通信機能が提供され，ノード間の通信によりノード間での同期・非同期タスクの実行を可能としている．実際のロボットとの連携動作では，図 1 に示すように制御用のホスト上で動作する“マスター”と各ノードがロボットや各ホスト上で実行される．

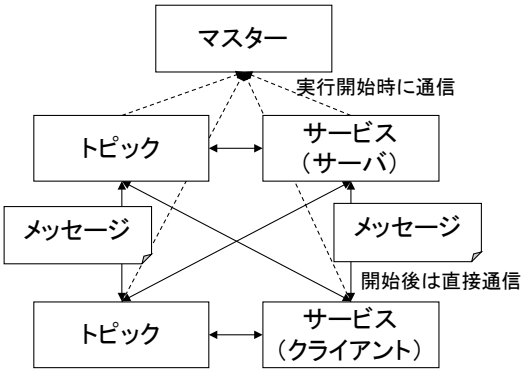

図 1　ROS におけるノード間通信の概観

以降では，分類予測を行うための学習モデルの構築と予測ラベル付与を行う分類学習タスクを各タスクに分割し，ROS 環境で実行するための入出力のオブジェクトの実装について説明する．さらに，Weka[5]に実装された分類学習タスクを実行する各種機械学習アルゴリズムを対象に ROS モジュールの実装について述べる．

2.1 機械学習アルゴリズムのタスク分割

本研究では，ROS 環境において機械学習アルゴリズムによるタスクを実行するため，既存の機械学習アルゴリズムの実行過程を入出力，および参照されるオブジェクトに着目し，整理した．図 2 は，分類や識別のタスクを機械学習による学習モデルを用いて実行するアルゴリズムから同定した「分類学習モデル生成タスク」と「分類予測ラベル付与タスク」の概観である．

「分類学習モデル生成タスク」では，訓練データ集合を入力として，学習パラメータを参照して分類予測を行うための式，決定木，if-then ルール集合，ネットワークなどの学習モデルを出力とするタスクとして定義する．「分類予測ラベル付与タスク」は，テストデータ集合を入力として，すでに生成されている任意の学習モデルを利用して，分類予測ラベルをテストデータ集合の各データに付与するタスクとして定義する．

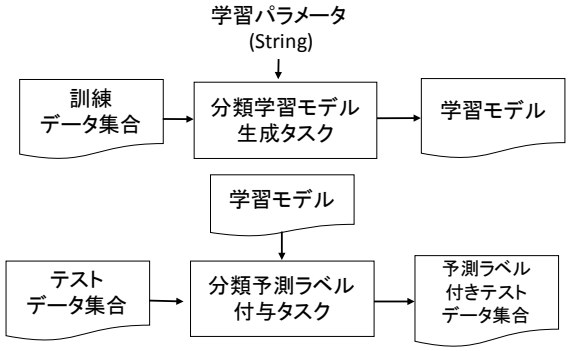

図 2 分類学習タスクの分割

以上に述べたタスク分割は，機械学習アルゴリズムを熟知した開発者には周知の事実であるが，タスク定義を明示することにより各モジュールを統一した入出力・参照を用いて記述することが可能となる．

2.2 分類学習タスクを実行する ROS モジュールの実装

ROS 環境では，各ノードの入出力を“メッセージ”と呼ばれる構造体として定義できる．機械学習アルゴリズムによる分類学習タスクにおいては，入出力及び参照となるデータ集合と学習モデルを図 3 に示すように定義した．分類学習タスクを実行する各 ROS サービスでは，受け取ったメッセージを解析し，必要な入力データ集合，学習モデルを取得し，プログラムを実行する．同様に実行結果については，メッセージを作成し，学習モデルおよびデータ集合を出力として返信する．ここで定義したメッセージでは，type に示されたオブジェクトの形式に従い，file であれば，file_name に示されたパスにあるファイルを用い，uri であれば，uri に示された URI からオブジェクトを取得し，str であれば，メッセージとして通信する文字列を str から入力として取得あるいは出力として送出する．

変数	ROSデータ型
type	string
file_name	string
uri	string
str	string

図 3 機械学習関連オブジェクトのメッセージ(.msg)の定義

本稿におけるメッセージの実装では，以下のような粒度でデータ集合と学習モデルのオブジェクトを

表す.

- データ集合：ライブラリ共通の形式毎
- 学習モデル：機械学習アルゴリズム毎

分類学習タスクに関するモジュールは，サービスとして実装する．各タスクを記述するため，データ集合「訓練データ集合」「テストデータ集合」「分類ラベル付与済みデータ集合」の 3 種類を定義し，各タスクのサービスを図 4 のように記述する．

「分類学習モデル生成タスク」のサービス定義：

入力	出力
訓練データ集合	各学習アルゴリズムの出力モデル
パラメータ(String型)	実行出力(String型)

「分類予測ラベル付与タスク」のサービス定義：

入力	出力
テストデータ集合	ラベル付きデータ集合
各学習アルゴリズムの出力モデル	実行出力(String型)

図 4　「分類学習モデル生成タスク」「分類予測ラベル付与タスク」のサービス(.srv)定義

図 3 に定義したデータ集合および学習モデル，図 4 のサービスの定義に従い，我々は図 5 に示す機械学習アルゴリズムにより分類学習モデルを構築し，分類予測タスクを実行する ROS モジュールを実装した．

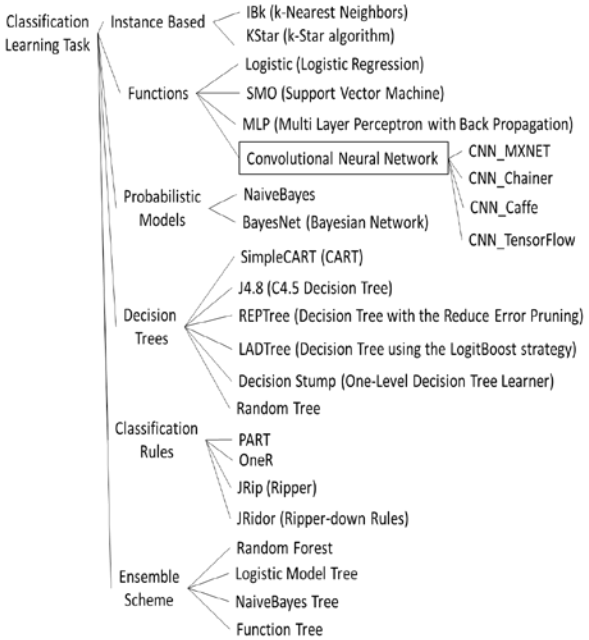

図 5 分類学習を行う機械学習アルゴリズム

3 節以降では，図 5 に示した末端ノードにあたる各アルゴリズムを Weka による実装を用いてサービスとして実装した．また，畳み込みニューラルネットワーク(CNN: Convolutional Neural Network)については，各末端ノードにあたる深層学習ライブラリを用いて実装した．

3 機械学習 ROS モジュールの実行と評価

本節では，2 章で示したタスク分割に従い実装した分類学習アルゴリズムを実行する ROS モジュールを用いて，一般画像認識タスクの各カテゴリ間の認識精度と ROS 環境における実行性能について評価実験を行い，その結果を示す．

各カテゴリ間の分類予測は，1 対 1 のカテゴリ同士の識別問題とし，精度の評価指標としては 10 回交差検証による平均正解率を用いる．

3.1 一般画像認識ベンチマークデータの前処理

本実験では，画像による一般物体認識タスクのベンチマークデータとして広く用いられる Caltech101[6]を利用し，各画像が属するカテゴリの認識精度の評価を行う．Caltech101 は，101 カテゴリの各カテゴリについて，30 枚～800 枚程度の JPEG 画像が収録されている．

本実験では，図 6 に示すように画像が属するカテゴリの分類予測精度について交差検証を行うため，訓練データ集合となる画像集合に対して特徴点抽出を行い，これらの特徴点をクラスタリングしてビジュアルワード辞書を構築する．さらに，Bag-of-VisualWords(BoVW)モデルによる各画像のビジュアルワードの出現頻度を特徴量として，表形式の訓練データ集合，および評価用のテストデータ集合を生成した．

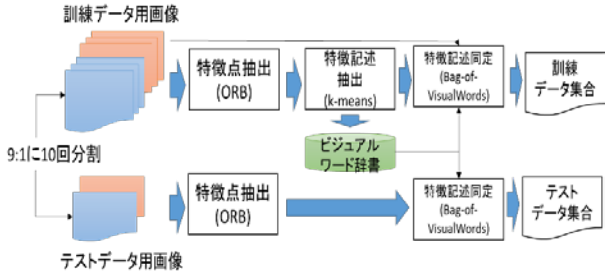

図 6 Bag-of-VisualWords による 10 回交差検証用訓練・テストデータ集合生成プロセスの概観

本実験では，特徴点抽出アルゴリズムとしてOpenCV3.1 に実装された ORB[7], クラスタリングアルゴリズムとして scikit-learn に実装された k-means(k=256)を用い，256 次元の BoVW データ集合とした．

3.2 代表的な機械学習アルゴリズムによる一般物体認識タスク

本実験では，30 枚～40 枚の画像が収録される 17 カテゴリについて，各カテゴリ間での分類予測の精度を行う．図 6 に示すように，2 つのカテゴリ間での認識精度を 10 回交差検証によって評価する．認識精度については，17 カテゴリ間の 136 ペアをクラスとした各データ集合について 10 回交差検証を行う．

評価実験おいてには，下記の代表的な 6 種類の機械学習アルゴリズムを ROS モジュール（サービスノード）として実行した．各モジュールは Python による実装され，Weka に Java により実装された各アルゴリズムは Python からのコマンド呼び出しを用いて実行される．

- 事例ベース学習：k-NN
- 決定木：J4.8
- ベイズモデル学習：Naïve Bayes
- サポートベクターマシン：SMO
- 従来型(3 層)ニューラルネットワーク：MLP
- アンサンブル学習：Random Forest

表 1 に各アルゴリズムを実行する ROS モジュールによるカテゴリ間認識精度の全平均正解率，各タスクの平均実行時間を示す．ROS モジュールの実行環境は，Xeon E5-2609 を搭載したサーバ(CentOS6.8)上に Docker を用いて構築した ROS 環境(Ubuntu 14.04LTS + Indigo)である．なお，ROS のマスターノードおよび各サービスノードは 1 つの Docker コンテナ上で動作させた．

表 1 の結果から，平均正解率においては，Random Forest が 88.99%と最も高い値を示している．さらに，Random Forest では，より短い学習モデル生成タスクの実行時間で，高い平均正解率が得られる結果となった．

表 1 代表的な機械学習アルゴリズムによる ROS モジュールの実行結果.

アルゴリズム (主要パラメータ値)	平均正解率(%) ±S.D.	平均学習モデル構築時間(秒)	平均分類予測ラベル付与時間(秒)
k-NN(k=3)	76.52 ±10.51	0.46 (0.07)	0.91
J4.8	77.79 ±12.55	0.52 (0.10)	0.56
Naïve Bayes	71.25 ±12.01	0.64 (0.06)	1.06
SMO	71.69 ±12.00	0.55 (0.10)	0.91
MLP (epoch=100)	82.59 ± 9.45	23.67 (22.64)	1.99
Random Forest (n=100)	88.99 ± 6.95	0.88 (0.34)	1.44

※カッコ内はWeka実行コマンド内でのモデル構築時間

実行時間については，分類学習モデル生成タスクにおいて，学習モデル構築時間（()内）と比べてタスク実行時間が 0.3～1 秒程度余分にかかっている．今回は，学習モデルの出力メッセージを type=”file”としてファイルシステム上に行ったため，書き込みに時間が余分にかかったものと考えられる．また，同様に分類予測ラベル付与タスクの実行についても，分類予測ラベルを付与したデータ集合の保存をファイルシステム上に行っているため，実行時間がかかっているものと考えらえるため，実サービスに取り入れるためには Autolib を利用して，実装を改善する必要があると考えられる．

4 畳み込みニューラルネットワークとの比較

本節では，分類や識別を行う畳み込みニューラルネットワークの構築と分類予測精度について，3 章で実行した従来型の機械学習アルゴリズムによるものとの比較を行う．

さらに，入出力の扱いやモジュール作成のコストに着目し，従来型の機械学習ライブラリを利用したモジュール作成と深層学習ライブラリとの相違について考察する．

4.1 画像カテゴリ間の認識精度の評価

本実験では Chainer[8]による畳み込みニューラルネットワークを利用し，画像の属するカテゴリ間の識別精度の評価を行う．畳み込みニューラルネット

ワークは, 画像識別をタスクとする事例[1]を参考に, 2 層の畳み込み層とプーリング層を設定し，その後に ReLU を活性化関数としたノードを配した層を 2 層用いる構造とした．この畳み込みニューラルネットワークを Chainer(1.20.1)を用いて Python(2.7)により実装し，訓練データ集合による畳み込みニューラルネットワークの学習タスクとテストデータ集合による分類予測タスクを実行する．

表 2 に 3.2 節で述べた 17 カテゴリ間での画像を入力とした 10 回交差検証による全体の識別精度（平均正解率）と各タスクの平均実行時間を示す．実行に用いた環境は，3.2 に示す環境と同一であるが，ROS モジュールとしては実行していない．また，Chainer は CUDA による GPGPU の利用も可能であるが，本実験では利用していない．

表 2 畳み込みニューラルネットワークによる画像カテゴリ認識タスクの実行結果.

アルゴリズム	平均正解率(%)±S.D.	平均学習モデル構築時間(秒)	平均分類予測ラベル付与時間(秒)
CNN_Chainer (epoch=20)	93.51 ±6.50	17.42	0.14

表 2 に示すように，畳み込みニューラルネットワークによる画像認識タスクの分類精度は，平均正解率において，BoVW による特徴量抽出と従来型機械学習アルゴリズムの組み合わせによるものと比較して高い性能を示している. 学習モデルの構築時間は, 従来型の 3 層ニューラルネットワークと比較して，少し短い程度ではあるが，BoVW による特徴量抽出に相当する処理 [2]を畳み込み層で行っていることを勘案すると圧倒的に短い実行時間で高い平均正解率を達成している．

また，予測タスクでは，ROS モジュールとして実装した従来型の機械学習アルゴリズムより短いが，ラベルを付与したデータ集合のファイルシステム上への出力を行っていないことから，ROS モジュールとした場合にはより実行時間がかかるものと考えらえられる．

4.2 従来型機械学習アルゴリズムライブラリと深層学習ライブラリの比較

本研究では，ROS モジュールとして機械学習アルゴリズムによるタスクを同定し，モジュールの実装を行っているが，各ライブラリを用いる際にはそれぞれ相違が生じる．

表 3 に入出力に着目した各ライブラリの特徴について，比較を示す.

表 3 従来型機械学習アルゴリズムを実装したライブラリと深層学習ライブラリの比較.

比較項目	従来型 機械学習 ライブラリ	深層学習 ライブラリ
入力データ	表形式データ, カラム指向データ	生データ （画像, テキストなど）
入力データの内部表現への変換	不要	必要 （ライブラリごと作成要）
入力データの形式	ライブラリ内では共通	ライブラリ間で異なる
特徴抽出	必要 （ライブラリが異なっても処理内容は共通）	不要
特徴抽出の処理内容	明示的	暗黙的
学習モデルの構造	一定	問題ごとに 最適化が必要
学習モデル構築パラメータの影響	既知なことが多い	未知なことが多い

表 3 に示したように，図 5 で畳み込みニューラルネットワークの実行環境として示した深層学習ライブラリなどの多くでは，入力の生データを内部データ表現に変換するプログラムを作成する必要がある. これは，様々な入力データへの対応としては柔軟性を確保できるが，従来型の機械学習アルゴリズムを実行する機械学習ライブラリと比較すると利用に際してのコストが高い．また，深層学習ライブラリ間

[1] 深層学習でアニメ顔を分類する with Chainer：http://qiita.com/hogefugabar/items/312707a09d29632e7288

[2] 今回の BoVW による特徴量抽出では，平均して約 8 分程度の処理時間を要した.

で共通した入力データの指定を行う形式は存在せず，各ライブラリを利用するために開発者が学習に要する時間が必要となる．

一方，従来型の機械学習ライブラリは表形式のデータ集合を用意することが多く，入力とする画像データなどの生データを 3.1 節で述べたように特徴抽出や特徴構築と呼ばれる前処理を行う必要がある．従来型の機械学習ライブラリでは，抽出した特徴に基づいてデータ集合を構築する際に人の専門家との対話が可能であるが，適切な特徴集合の作成には困難が伴う．

学習モデルを構築するためのパラメータに関しては，従来型の機械学習アルゴリズムでは性能に影響を与える範囲がアルゴリズムの理論的定義から説明がある程度可能である．しかし，深層学習では，各問題に対して設定するパラメータが複雑であり，深層ニューラルネットワークの構造を含めて，最適な値や構造の設定を試行錯誤で行う必要がある．これは，「ハイパーパラメータ決定問題」と呼ばれ，適切な学習パラメータやネットワーク構造の設定には，多くの分析事例についての経験と各深層ニューラルネットワークの構築理論に精通する必要がある．4.1 に示した畳み込みニューラルネットワークに関しても，今回の小規模な画像認識タスクでは表 2 に示すように高い精度を達成することができたが，他の画像認識タスクでも同様の結果が得られるとは限らない．

5 おわりに

本研究では，ロボットオペレーションを行うミドルウェアである ROS 上で動作する機械学習モジュールについて，分類学習モデル生成タスクと分類予測ラベル付与タスクに分割する方法と実装を行った．

本稿では，代表的な機械学習アルゴリズムを Weka より利用したモジュール群による一般物体認識問題からの画像を用いた分類予測精度の評価，および畳み込みニューラルネットワークとの比較について考察した．

今後の課題として，本研究において実装した機械学習アルゴリズムの ROS モジュールを用いて，より円滑に対人サービスの業務プロセスを実行するためのプロセス記述を行い，実サービスでの実証を行っていくことを予定している．

謝辞

本研究の一部は，JSPS 科研費 JP 26240036，および JST/CREST「実践知能アプリケーション構築フレームワーク PRINTEPS の開発と社会実践」の補助によるものである．

参考文献

[1] ROS: http://www.ros.org/

[2] 山口高平，中野有紀子，斎藤英雄，森田武史，青木義満，萩原将文，斎藤俊太 : 知能共進化のための実践知能アプリケーションプラットフォーム PRINTEPS, 人工知能学会全国大会論文集， Vol. 29, pp.1-3, http://printeps.org/ (2015)

[3] M. Tenorth and M. Beetz: KnowRob: A knowledge processing infrastructure for cognition-enabled robots, The International Journal of Robotics Research, Vol. 32, No. 5, pp. 566-590 (2013)

[4] 阿部秀尚，森田武史，山口高平 : ROS 環境上での機械学習実行モジュールの設計と実装，第 79 回情報処理学会全国大会 (2017) （発表予定）

[5] E. Frank, M. A. Hall, and I. H. Witten, The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, (2016)

[6] Caltech101: https://www.vision.caltech.edu/Image_Datasets/Caltech101/

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski: ORB: An Efficient Alternative to SIFT or SURF, Proc. Of International Conference on Computer Vision, pp. pp. 2564-2571 (2011)

[8] Chainer: http://chainer.org/