機械学習のメタ学習機構のための知識記述と実行環境の検討

A Study of Knowledge Representation and Execution Environment for Meta-Learning Scheme of Machine Learning Algorithms

阿部秀尚 [1]

Hidenao Abe[1]

[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

Abstract: Recent years, “Meta-Learning for machine learning” method have been gradually popularized to win competitions for machine learning accurarices for given dataset such as Kaggle and OpenML. These competitions are aimed to encourage for educating superior experts for machine learning and data science fields. However, in order to determine hyper-parameters for applied machine learning algorithms in such competitions, some participations often use the meta-learning tools for automating the process for selecting the best machine learning algorithms and the parameters. This spoils the efforts to learn knowledge about to choice adequate machine learning algorithms and other pre-processing process for each given dataset, because these tools output just selected process, called pipeline, and hyper-parameters for the selected one rather than describing the process and the reason for the selection. In this paper, a framework for selecting proper process for given dataset and problem with describing knowledge about the pipelines is discussed as another meta-learning.

はじめに

機械学習を利用した識別・判別のタスクを様々なデータに対して，当該タスクのために十分高い精度が得られるようにするため，適切な機械学習アルゴリズムの選定が必要となる．このような，機械学習アルゴリズム選定は，「メタ学習」[1]と呼ばれ，アルゴリズム選定を支援する手法が開発されてきた．

一方, 近年, Kaggle[2]や OpenML[3]などのように，提供されたデータ集合に対して精度の高い機械学習アルゴリズム選定を行うコンペティションが広く認知されるようになってきている[4]．従来，機械学習アルゴリズム選定を競うコンペティションは，国際会議の併設ワークショップとして開催されてきた．しかし，前述のような常設のコンペティションプラットフォームの登場により，現在は，機械学習エンジニアの能力証明としても利用されるようになってきている．

しかしながら，どのような入力データに対して，どのような機械学習アルゴリズムを選定すればよいか，という問題を解決する過程は，明示的な知識記述による問題として扱われていない．このため，従来のメタ学習機構は，実行可能なソースコードやコマンドを出力することが可能であるが，仕様探索の過程がアルゴリズムやパラメータの選択によるものなのか，実装に依存した影響なのかを判別することが困難であり，仕様探索過程が機械学習を扱うエンジニアにアルゴリズムやパラメータの選定に関する知識の蓄積に寄与していないのが現状である．

本稿では，上述の状況に対し，機械学習における識別・判別を行うタスクを実行するアルゴリズムおよびそれらを含む一連の処理を対象に，先行研究[5][6]を踏まえた明示的な処理単位の記述と仕様空間の探索を可能とする知識記述について検討する．次に，知識記述に基づく機械学習アルゴリズムとそれらのパラメータを仕様記述とし，一連の処理の評価指標に基づく評価の向上を目指した仕様空間の探索と一連の処理の実装との連携について検討する．

メタ学習機構に関する関連研究

メタ学習機構は，所与のデータセットに対する適

切な機械学習アルゴリズムの選定において，データセットの特徴量[7]などを基に自動的に機械学習アルゴリズムなどのデータ分析手法とそれらのパラメータを選択する枠組みである[8].

機械学習アルゴリズム選定のためのメタ学習ツール

これまで, 所与のデータセットと問題設定に対し，機械学習アルゴリズムを含む処理全体（以降,「パイプライン」と呼ぶ）を選定するため，いくつかのメタ学習機構が開発されてきた．問題設定の多くは，分類予測のタスクが用いられ，パイプラインの評価の対象の多くは正解率や F1-Score, ROC 曲線の AUC といった学習モデルの正確さ（精度）に関する評価指標が用いられる．

CAMLET[5]は，分類学習に用いられる機械学習アルゴリズム，分類器洗練化手法などを基にメソッドリポジトリという機能単位を体系化し，所与のデータセットに最適な帰納アプリケーションと呼ぶ機械学習アルゴリズムを再構成するメタ学習機構である．阿部は，CAMLET を基に属性選択アルゴリズムの再構成と選定を行う機構を実装し，平均して最も高い正解率が得られる属性選択アルゴリズムと同等の正解率がより少ない計算量で得られることを示した[6][1].

Auto-Weka[9]は，Java で実装された機械学習ライブラリである Weka[10]を利用して，属性選択アルゴリズムの適用から機械学習アルゴリズムに至る処理を所与のデータセットに対してパラメータに至るまで最適化することが可能なツールである. ユーザは，所与のデータセットに対し, Weka の実行コマンドとして最適とされた属性選択アルゴリズム（不要であれば，選定されない）と機械学習アルゴリズムを分類予測タスクのパイプラインとして得ることができる．

Auto-sklearn[11]は，Python により実装された機械学習ライブラリである scikit-Learn[12]を用いて，所与のデータセットに最適なパイプラインを実行可能な形（sklearn.pipeline により実行可能なコマンド）として得られるメタ学習機構である．

このほか，TPOT[13]，ATM[14]，AutoCompete[15]など scikit-Learn に基づく機械学習アルゴリズムの実行環境としたメタ学習機構が開発されている．

以上のように，現在，良く利用されるメタ学習機構では，それぞれのプログラミング言語や実行環境が提供するライブラリに依存した実装となっている．これは，再現性を確認するために実装レベルでの実現には寄与するが，実装とは独立した選定根拠を見出すことは困難である．さらに，近年，機械学習アルゴリズムコンペティション向けに開発されるツールは，正解率などの評価指標において上位のパイプラインをアンサンブル学習させ，さらに複雑なパイプラインを構成するなど，実行されるパイプライン自体を理解することが困難になってきている．

機械学習アルゴリズム選定のための仕様空間探索法

所与のデータセットに対する機械学習アルゴリズムの選定およびパイプラインの最適化のため，各メタ学習手法は, 実装された各処理単位を組み合わせ, 評価指標が最も高くなるよう試行錯誤を繰り返し実行する．処理単位の組み合わせ数は，実数パラメータを含むと無限に近い数となるが，概ね有限な組み合わせの中からの探索（以降，仕様空間探索と呼ぶ）となる．

仕様空間探索のため，機械学習アルゴリズムの適用プロセス，あるいはパイプラインをルートとし，各処理の組み合わせ，パラメータの組み合わせを木構造の展開として定義し, 探索の対象としている（図 1）．探索手法は，木構造展開を確率的に行う手法[9][11]と遺伝的アルゴリズムによる処理単位およびパラメータの組み合わせを探索する手法[13]が多く採用される．このような木構造は，後述するパイプラインの仕様記述を JSON や YAML, XML といった機械可読型の書式とする場合との相性も良いと考えられる．

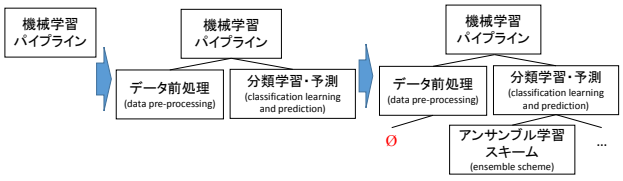

図 1 : 木構造の展開によるパイプラインの作成過程の概観

機械学習タスクと機械学習モデルの出力形式

機械学習アルゴリズムによる分類予測問題を対象としたタスクは，図 2 に示すように，所与の訓練データセットを利用した「分類学習モデルの生成タスク」とテストデータセットに対する「分類予測ラベ

[1] しかしながら，公開ツールとしていないため，現在の機械学習コンペティションでは利用されていない．

ルの付与タスク」に大別される[13].

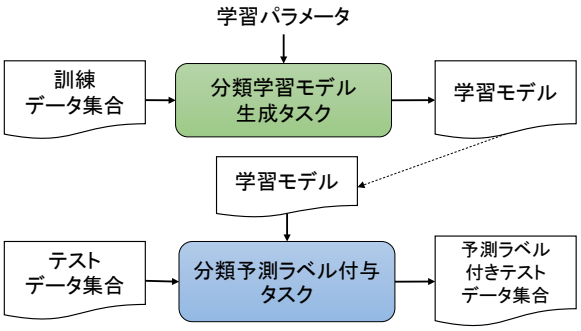

図 2：分類予測問題を対象とした機械学習タスク

分類学習モデルの出力は，機械学習ライブラリに依存した形式との共通の形式のファイルが用いられる．Weka を用いた場合は，シリアライズされた Java オブジェクトが出力される．scikit-learn を用いた場合は，Python オブジェクトのシリアライズが出力される．

これらのライブラリでは，分類予測ラベルの付与を行う際，各ライブラリに依存するシリアライズされた分類モデルのほか，実装言語に依存しない PMML[17]などの形式を利用することが可能である．

構成的メタ学習機構のためのプロセスと機械学習メソッドの知識記述

機械学習アルゴリズムを含む，一連の処理であるパイプラインは，各処理単位（メソッド）の連なりから構成される．これらのメソッドを体系化し，プロセス記述に提供することで，一連の処理の知識記述を可能とする．

図 3 は，一連の処理内で処理の対象となるデータセットについて，代表的な対象（Object）を体系化したものである．階層関係は is-a または subClassOf の関係を利用しており，末端に行くほど具体的なデータとなり，メソッドの入力(input)・出力(output)・参照(reference)の値として記述する．

一連の処理を構成する代表的なタスクは,「データ前処理」「学習モデル生成」「学習モデル洗練化」「学習モデルによる特徴追加」「(パイプライン全体での)評価」であり，これらの各タスクの中で他のタスクを逐次的に並べ，メソッドによっては入れ子とし，パイプラインの一連の処理を構成する．機械学習アルゴリズムやデータマイニングの適用に当たっては，これらのメソッドを適宜組み合わせ，所与の問題解決にあたっているため，これらのメソッドを自動的に再構成するためには機械可読型の記述とする必要がある．

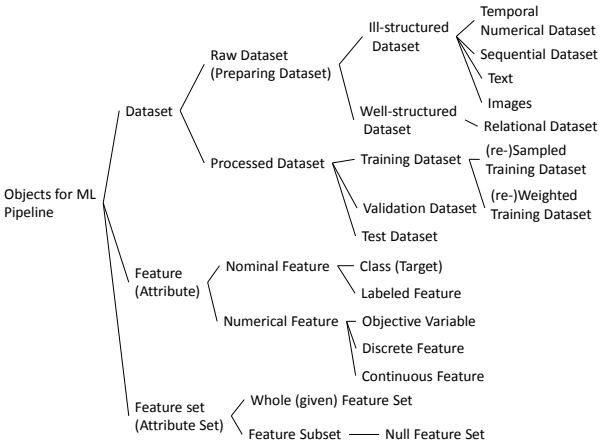

図 3：データの前処理タスクで処理対象となる代表的なオブジェクトの体系化

以下に機械学習アルゴリズムを含む一連のパイプラインでの代表的なタスクを列挙する．

- データ前処理
  - 特徴（属性）変換
  - 特徴（属性）構築
  - 特徴（属性）選択
  - 事例選択（サンプリング）
- モデル生成
  - 分類学習モデル生成
  - クラスタリング
  - パターン生成
- 学習モデル洗練化
  - 訓練データ洗練化（更新）
  - 学習モデル洗練化
- 学習モデルによる特徴追加
  - ラベル付与
  - 特徴（属性）追加
- 評価
  - モデルの評価
  - パイプライン処理の評価

機械学習アルゴリズムを含む一連のパイプラインは，データマニングプロセスにおいて,「データの前処理」とされる“データの変換”，“データの補強”，“データのコード化”のほか,“属性選択”,“事例選択”などがある [2]． Weka では，これらの処理は weka.filters パッケージに多くが実装されているが，

[2] 機械学習コンペティションなどでは，発見的な特徴量の追加や合成などが行われることもあり,「特徴量エンジニアリング」と呼ばれている．

scikit-learn では一部にとどまっているため，実装とは独立して処理を列挙し，体系化する必要がある．データの前処理の一部である「特徴選択」については，[6]においてメソッドを体系化したメソッドリポジトリについて述べられている．

「データの前処理」を経たデータセットは，分類学習モデルやクラスタ，特徴的なアイテム集合や系列パターンといったモデルを生成する．これらのタスクで用いられるメソッドは，機械学習アルゴリズムなどのモデル生成アルゴリズム開発の中心であるため，非常に多くの具体的な手法が存在する．例えば，「分類学習モデル生成」タスクに関しては，[13]において，図 4 のように体系化を行った．

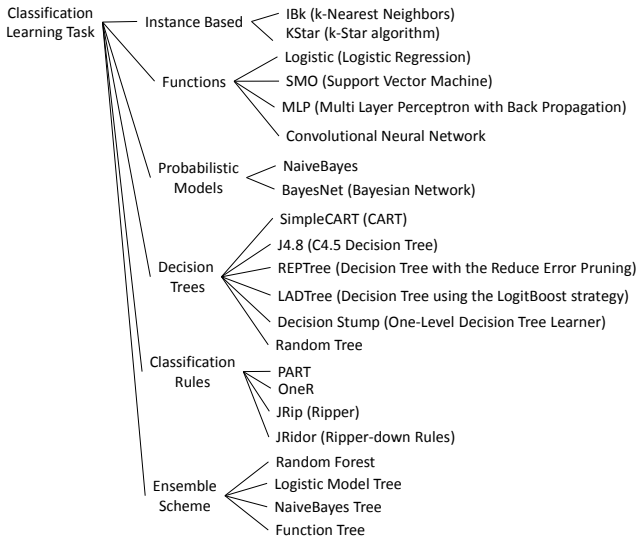

図 4 :「分類モデル生成タスク」で処理を実行する各メソッドの体系化（[13]掲載図を改変）

以上のように，各タスクにおいても，それぞれの具体的な処理単位であるメソッドと対応するレベルまで詳細化したメソッドリポジトリを構築し，一連の処理をパイプラインとして展開した時に実装と対応付けを可能とする．

構成的メタ学習機構のためのプロセス知識記述と実行環境

機械学習アルゴリズムの実行を含むパイプライン処理の一連のプロセスは，図 1 に示したように木構造の展開として扱うことが探索問題としての扱いも容易である．このようなメタ学習機構の動作プロセスは，[9][11][13-15]に示されており，探索問題としての扱いの良さが反映されているものと考えられる．

本研究でも先行研究に倣い，図 5 に示す記述を用い，展開可能なキーの値をメソッドオブジェクトに置き換えていくことにより，各処理単位であるメソッドを組み合わせ可能にする．展開の際は，各メソッドに記述された前後のメソッドの入出力の制約に従い，パイプラインを再構成する．このため，前節に示した各タスクの記述を上位のメソッドリポジトリ（プロセスメソッドリポジトリ）として構築し，各タスクの展開段階では，それぞれのメソッドリポジトリの記述を参照して，具体的な処理プロセスの構築を行っていく．各メソッドの記述を JSON でのオブジェクトとした場合，プロパティ（属性）については，type, label, input, output, reference, parent（親タスク，親メソッド）, e_parameters（展開すべきタスク，メソッドを含むパラメータ）, c_parameters（値を決定すべきパラメータ）をはじめとしたタスクとメソッドの定義を値として，記述する．

```
{ “label”: “classification learning”,
  “type”: “task”,
  “parent”: “machine learning pipeline”,
  “input”: “training dataset”,
  “output”: “classification learning models”,
  “e_paramters”: [ “instance based learning”,
                   ...,
                   “ensemble scheme”]
}
```

```
{ “label”: “decision trees”,
  “type”: “method”,
  “parent”: “classification learning task”,
  “input”: “training dataset”,
  “output”: “decision tree model”,
  “c_paramters”:[
    { “name”: “max_depth”,
      “value”: 3},
    { “name”: “pruning”,
      “value”: true},
    ...]
}
```

図 5 : JSON によるタスクおよびメソッドの記述例

知識記述上で作成した仕様記述は，図 6 に示す構成的なプロセス（構成的メタ学習）により，実装に依存しない機械可読型の記述形式でプロセスが記述され，各プログラミング言語の機械学習ライブラリなどを利用したソースコード，あるいはコマンド列に変換される．実行された情報は，洗練化機構によって新たなプロセス仕様記述が生成られ，実行を繰り返す．これにより，仕様の記述レベルと実装レベルを切り離すことができ，入力されるデータセットと問題設定に即した一連のパイプライン処理プロセスの構築に対して，ライブラリに依存した処理の有無による処理に関する知識獲得の機会の逸失などを防止する．

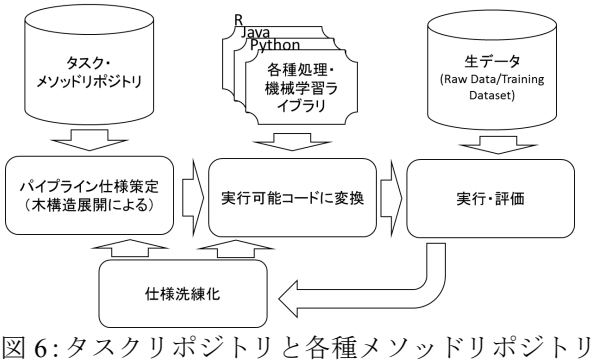

図 6 : タスクリポジトリと各種メソッドリポジトリ

を利用した構成的メタ学習プロセスの概観

さらに大規模な分散実行環境の構築にあたっては，Git・Gitlab のようなリモートリポジトリを有する CI (Continuous Integration)可能な環境を利用し，各パイプラインの仕様記述，あるいは各機械学習ライブラリ向けのソースコードを連続時に取り込んで実行することが考えられる．

おわりに

機械学習ライブラリでは，実装言語が異なっても同一の機械学習アルゴリズムや関連のフィルタ処理等の処理であれば，精度や動作は同一であるべきである．しかしながら，現状では，乱数発生過程の際などにより，同一データセットに対しても，ライブラリの違いによって精度に差異（時として統計的に有意と認められる差異）が生じることがある．本来，このような差異は生じるべきでは無いが，機械学習コンペティションでの各ライブラリによる実行により，顕在化することがみられるようになってきた．

このため，所与のデータセットに対する機械学習アルゴリズムを含む一連の処理を最適化する際に，機械学習ライブラリの選定が含まれる研究事例[18]も見られるが，本来は機械学習アルゴリズムとそれに関連する処理の選択がその本質である．

本稿での提案により，知識記述レベルでの機械学習アルゴリズムを含む一連のパイプラインの最適化である仕様探索と実行レベルでの各機械学習アルゴリズムや関連処理の実装が分離されて扱われることが望まれる．今後は，本稿での検討に基づくツールの開発と公開を行っていく [3].

謝辞

本研究の一部は，2019 年度文教大学学内競争的研究資金による．ここに記して感謝の意を表する．また，構成的メタ学習機構に関する研究は，慶應義塾大学理工学部 山口高平 教授はじめ，多くの研究者による先行研究，ご助言，ご指導による成果であり，ここに深く感謝いたします．

参考文献

[1] Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning:Applications to Data Mining. Springer-Verlag Berlin Heidelberg, (2009)

[2] Kaggle: https://www.kaggle.com/

[3] OpenML: https://www.openml.org/

[4] 馬場雪乃：機械学習コンペティションの進展と今後の展開(<特集>人工知能研究のベンチマークとは-標準問題・データセット・評価手法-), 人工知能，Vol 31, No. 2，pp.248-253, (2016)

[5] 酢山明弘，山口高平：オントロジーを利用した帰納アプリケーションの自動合成，人工知能学会誌，Vol. 15, No. 1, pp. 155-161, (2000)

[6] Abe, H., and Yamaguchi, T.: A constructive meta-level feature selection method based on method repositories. Journal of Computers, Vol. 1, No. 3, pp.20-26, (2006)

[7] Peng, Y., Flach, P. A., Soares, C., and Brazdil, P.: Improved dataset characterisation for meta-learning. In Proceedings of International Conference on Discovery Science, pp. 141-152, (2002)

[8] AutoML.org：https://www.automl.org/

[9] Kotthoff, L., Thornton, C., Hoos, H.H., Hutter, F., Leyton-Brown, K.: Auto-WEKA 2.0:Automatic model selection and hyperparameter optimization in WEKA. Journal of MachineLearning Research, Vol. 18, No. 25, pp.1–5, (2017)

[10] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA datamining software: An update. ACM SIGKDD Exploratians Newsletter, Vol. 11, No. 1, pp.10–18, (2009)

[11] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F.: Efficient and robust automated machine learning. In Advances in neural information processing systems (NIPS2015), pp. 2962-2970, (2015)

[12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher,M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. JMLR, Vol. 12, pp. 2825–2830, (2011)

[13] Olson, R.S., Bartley, N., Urbanowicz, R.J., Moore, J.H.: Evaluation of a tree-based pipeline optimization tool for automating data science. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO'16). pp. 485–492, (2016)

[14] Swearingen, T., Drevo, W., Cyphers, B., Cuesta-Infante, A., Ross, A., & Veeramachaneni, K.: ATM: A distributed, collaborative, scalable system for automated machine learning. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data 2017), pp. 151-162, (2017)

[3] https://github.com/hidenaoA/ConstructiveMetaLearning/

[１５] Thakur, A. and Krohn-Grimberghe, A.: Autocompete: A framework for machine learning competitions. In AutoML Workshop, International Conference on Machine Learning 2015, (2015)

[１６] 阿部秀尚，森田武史，山口高平：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，人工知能学会 第 110 回知識ベースシステム研究会，pp.18-23, (2017)

[１７] PMML 4.4 Standard: http://dmg.org/pmml/v4-4/GeneralStructure.html

[１８] Maher, M., and Sakr, S.: SmartML: A Meta Learning-Based Framework for Automated Selection and Hyperparameter Tuning for Machine Learning Algorithms. EDBT: 22nd International Conference on Extending Database Technology, 10.5441/002/edbt.2019.54. hal-02087414, (2019)