機械学習のメタ学習機構のための知識記述と実行環境に関する検討

文教大学情報学部情報システム学科

阿部秀尚

hidenao@shonan.bunkyo.ac.jp

研究背景

- 適切な機械学習アルゴリズムの選定に対する需要増大
  - センサデータの収集やデバイスの進化によるデータ収集
    →深層学習をはじめとする機械学習アルゴリズムの
    適用可能な場面の増加
    - いわゆる「AIブーム」→幻滅を危惧
    - データ収集から分析（モデル生成），活用までの知識を持つ人材が必要
  - 適切なデータの加工・処理，機械学習アルゴリズムの選定が行える人材育成
    →コンペに“勝つ”（主に高い正解率を得る）知識
    ＝臨機応変にデータ収集や機械学習を扱う知識？
    - ツールを正しい目的で使っているか，データの加工・処理（特徴量エンジニアリング）についての根拠説明が可能か，評価値に対して正しく理解しているか，など心配が尽きない

⇒ **断片的な経験による人材の育成が現状**

分類学習に関するメタ学習研究

「より高い正解率」を目指して開発されてきた手法

- アンサンブル学習
  - Stacking：D. H. Wolpert, “Stacked generalization” Neural Networks, vol. 5(2), pp.241-259 (1992)
  - Boosting：Y.Freund and R.E.Schapire, “Experiments with a New Boosting Algorithm”, Proc. of The 13th Int'l Conf. on Machine Learning, pp.148-156 (1996)
  - Bagging：L. Breiman, "Bagging Predictors", Machine Learning, vol.24, pp.123-140 (1996)
- データ特徴量に基づくアルゴリズム選択
  - DCT: Y. Peng, P. A. Flach, C. Soares, & P. Brazdil, “Improved dataset characterisation for meta-learning”, In Int’l Conf. on Discovery Science (DS2002), pp. 141-152 (2002)
- メソッドリポジトリに基づくメタ学習

←私がメタ学習の研究に関わり始める

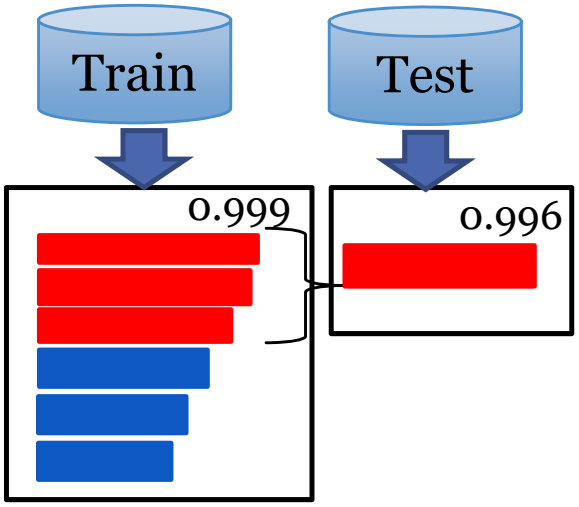
  - CAMLET: 酢山明弘，山口高平：オントロジーを利用した帰納アプリケーションの自動合成，人工知能学会誌，Vol. 15, No. 1, pp. 155-161, (2000)
  - 属性選択に対するメタ学習: H. Abe and T. Yamaguchi “A constructive meta-level feature selection method based on method repositories” Journal of Computers, Vol. 1, No. 3, pp.20-26, (2006)
- 機械学習ライブラリに基づくメタ学習
  - Auto-Weka: 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 Machine Learning Research, Vol. 18, No. 25, pp.1–5, (2017) ←Auto-Weka(1.0)はKDD 2013で発表
  - Auto-sklearn: 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)

機械学習コンペとメタ学習ツール

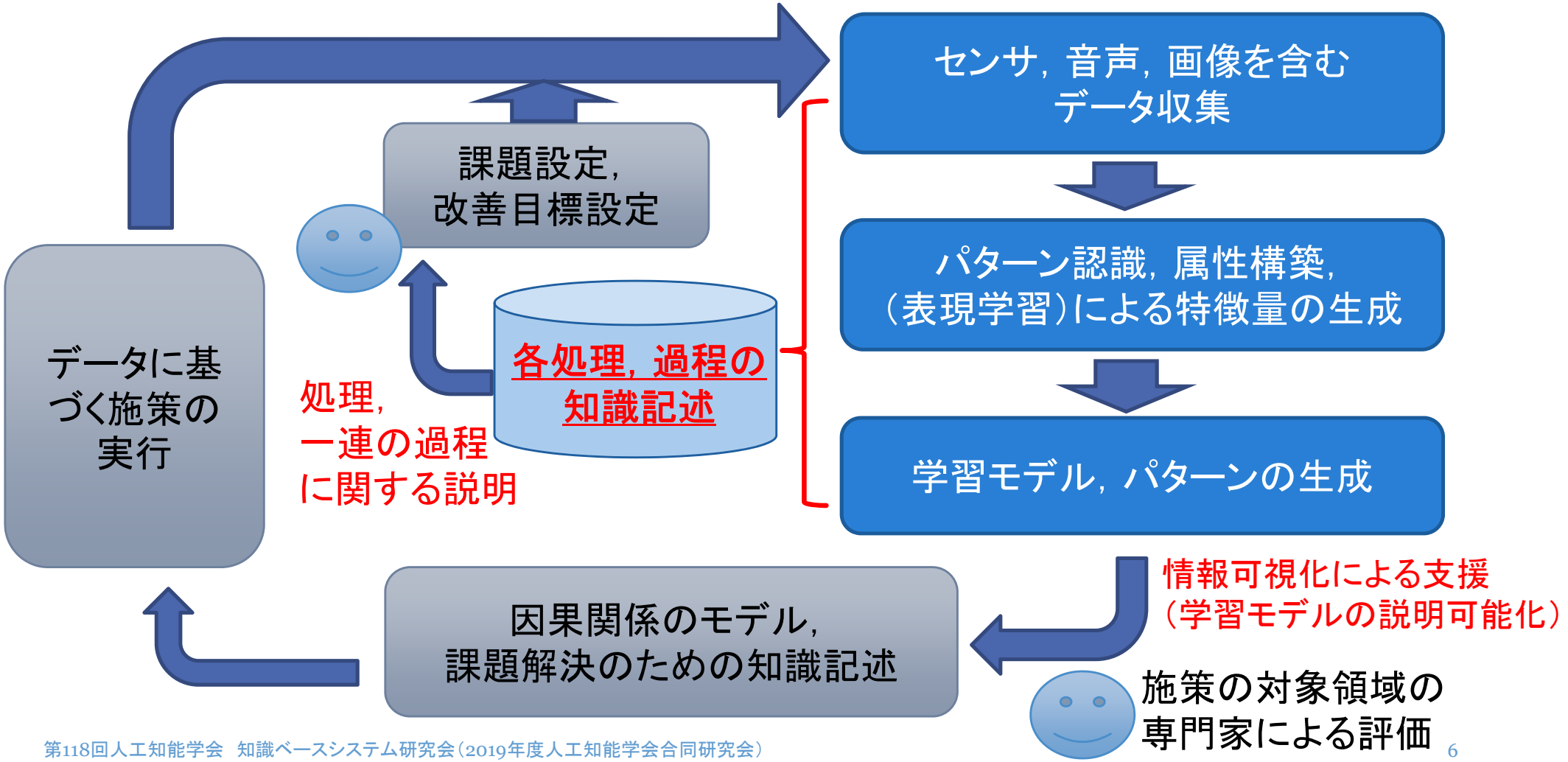
- 機械学習コンペ
  - 国際学会併設ワークショップ
    - KDD Cup: ACM SIGKDD主催のKDDに併設（1999～）
    - Discovery Challenge: ECML/PKDDに併設（1999~）
    - Data Challenge: IEEE ICDMに併設
  - 一般参加型コンペティション
    - **Kaggle**：一般企業などからのデータセットと問題の提供（企業としては2010～）
    - OpenML：古典的なベンチマークデータセット中心→一連の処理（パイプライン）の収集
- メタ学習ツール
  - 機械学習アルゴリズムの選定自動化・支援：TPOT，ATM，AutoCompeteなど
    - AutoMLというプロジェクトとして取りまとめが進められている（https://www.ml4aad.org/）
  - 機械学習アルゴリズムのパラメータ調整自動化：Optuna，GridSearch(scikit-learnの一機能)など

機械学習コンペと課題

- 機械学習コンペからの知見
  - 上位の解法を読み解くと様々な工夫がみられる
  - 工夫が一貫して効果を発揮するかどうかまでは不明
    - 再現性の問題が最近指摘されている
  - 最後はパラメータ調整
- 正解率向上の弊害
  - 非常に複雑なモデルが上位を占めることがある
    - 「自チーム内で実行したトップN位までのアルゴリズム(深層学習やアンサンブル学習によるものを含む)をさらにアンサンブルさせて最終結果を得た」といったものもある
  - 何が“工夫”として効いたかの根拠が「それで正解率向上に効いたから」となってしまいがち
    - 画像認識などではよいかもしれないが，企業の戦略策定のためのデータ分析では不適



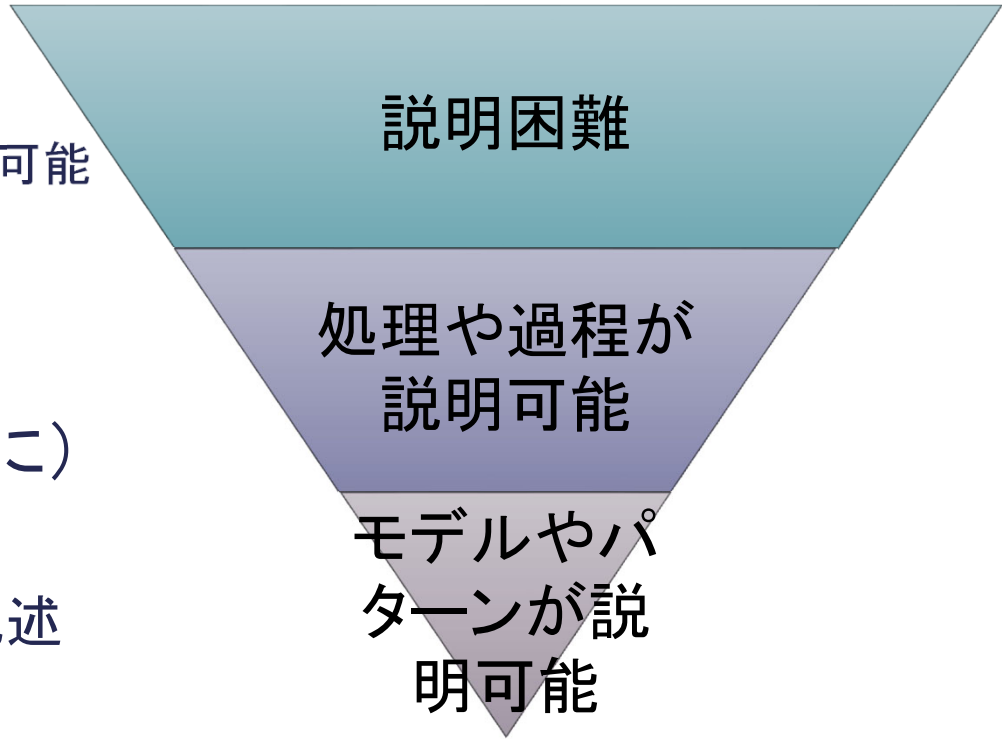
データに基づく施策実行プロセスと知識記述による説明可能化



機械学習(を伴う一連の処理)の説明可能性

XAI: Explainable Artificial Intelligence

- 説明可能性の段階
  1. 説明不可能
     - 例:ニューラルネットワークでのみモデル化可能
  2. 処理の過程が説明可能
  3. 学習モデルやパターンが説明可能
- 処理の過程を説明(共通理解可能に)するには
  - 入出力のデータやモデルを表す知識記述
  - 各処理(メソッド)の知識記述
  - 一連の処理の過程の知識記述



機械学習を伴う一連の処理のための知識記述

- 知的な作業を伴う処理（タスク→メソッド）のモデル化

図表 2-1 「基準」の構成要素 ＊

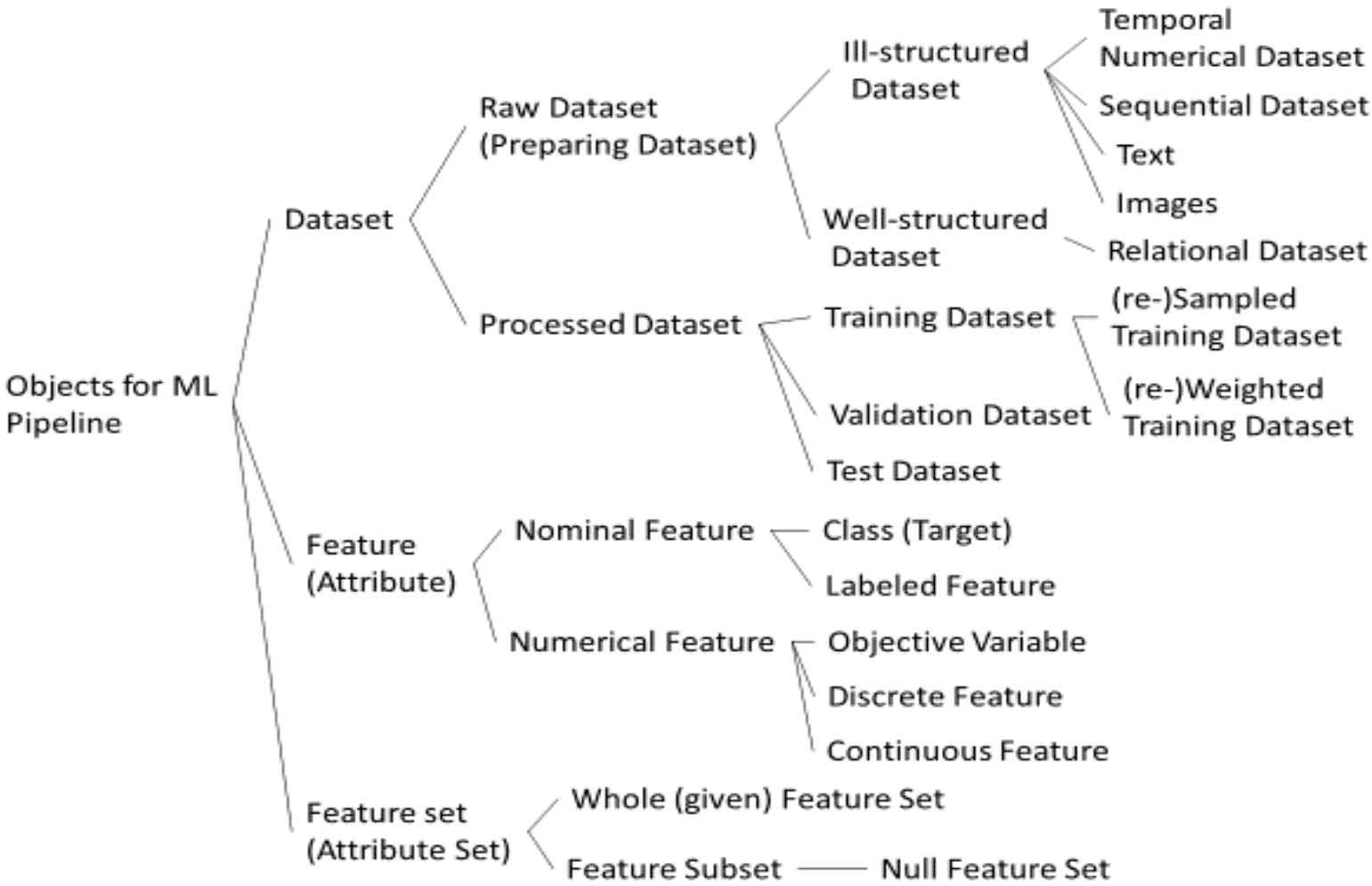
- 入出力のデータやモデルを表す知識記述
  - 体系化した入出力オブジェクトや学習モデル，パターンの記述
- 各処理（メソッド）の知識記述
  - 体系化したメソッドの記述
- 一連の処理の過程の知識記述
  - タスク，メソッド間の入出力オブジェクトに基づく一連の処理の記述

＊「人工知能技術の行政における活用に関する調査研究」報告書，一般社団法人行政情報システム研究所，2016.より

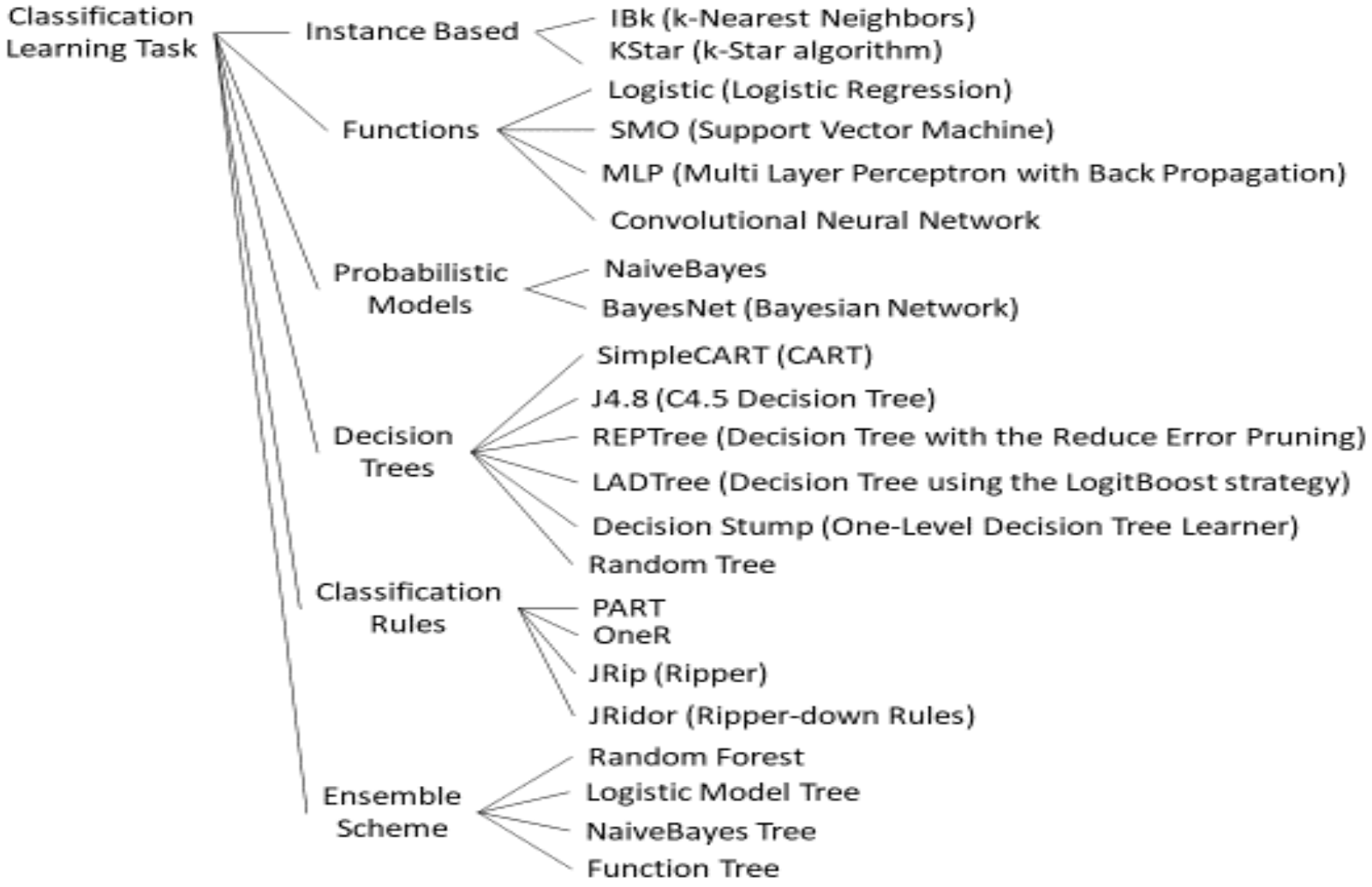
機械学習アルゴリズムの実行を伴う一連の処理で実行されるタスク ※代表的なタスク

- 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

データの前処理(属性構築)に関わるオブジェクトの体系化



分類学習タスクに関する メソッドの体系化

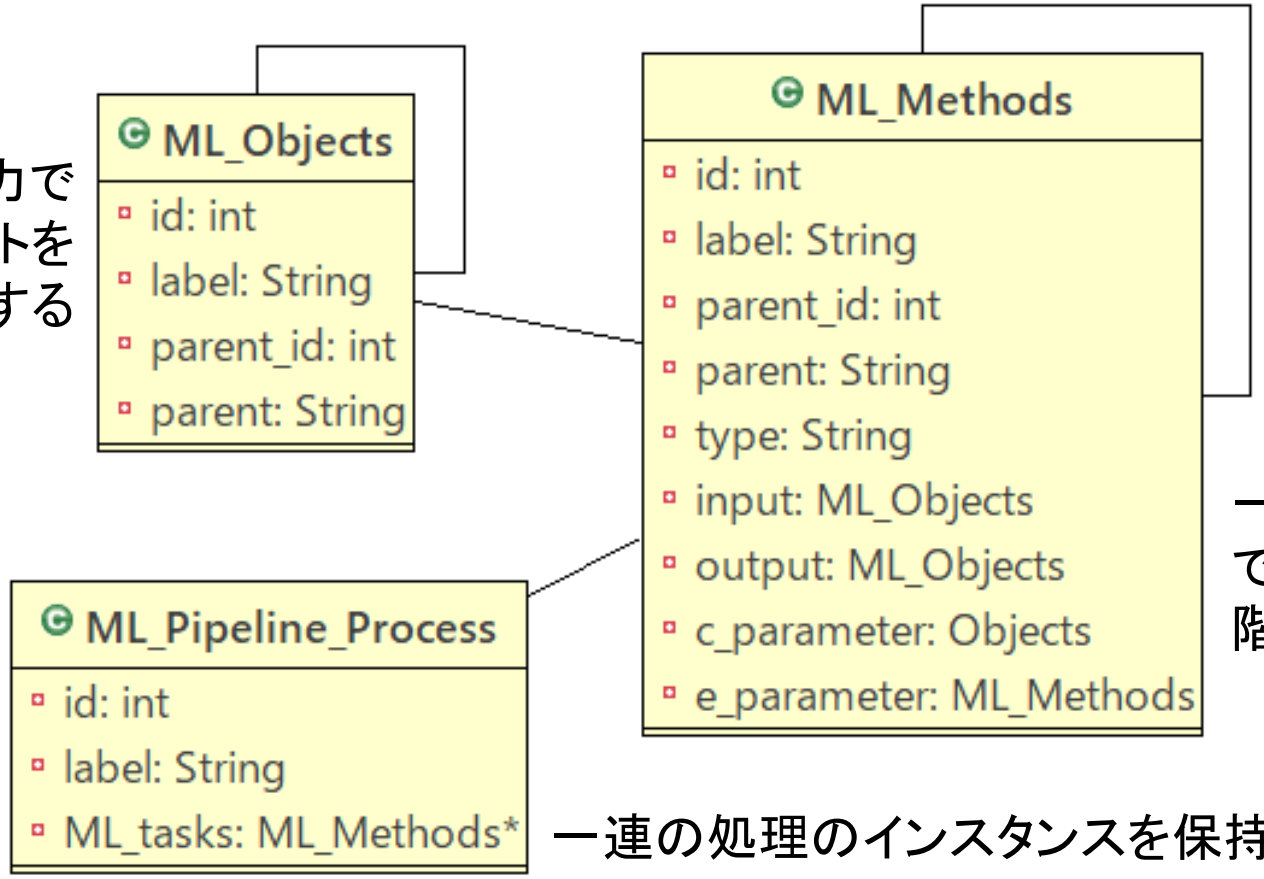


ライブラリの実装ノード，特異なアルゴリズムなどはさらに詳細化

メソッドリポジトリと機械学習オブジェクトの知識記述（データモデル）

※2019.11.22時点

一連の処理の入出力で扱われるオブジェクトを階層化して保持する



一連の処理の内容であるタスクとメソッドを階層化して保持する

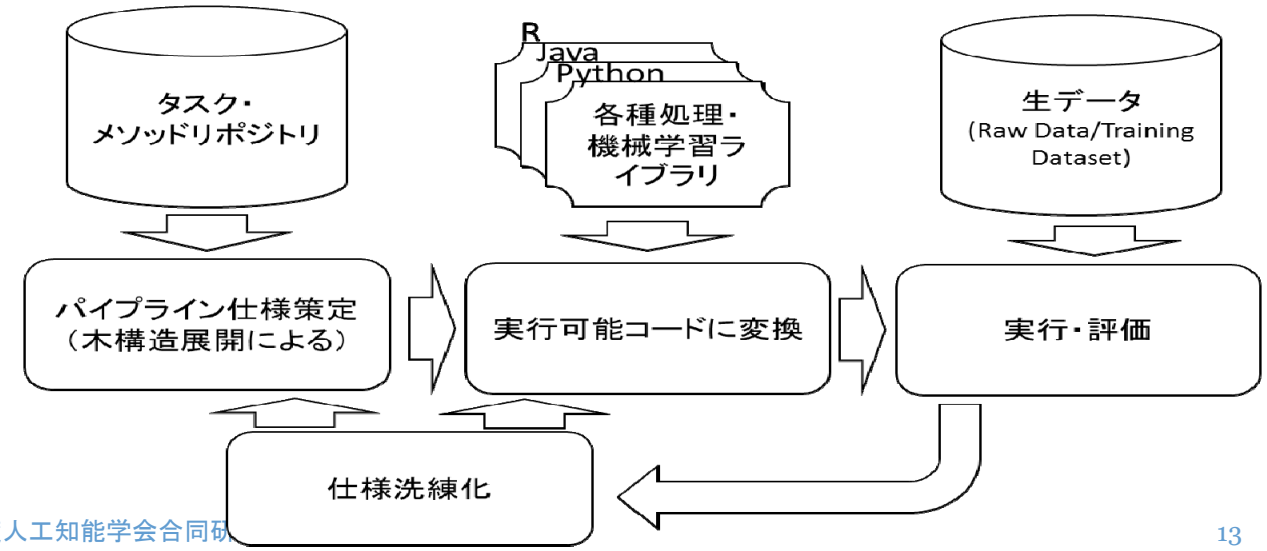
一連の処理のインスタンスを保持する

メソッドリポジトリに基づく メタ学習機構

機械学習ライブラリに基づくメタ学習機構

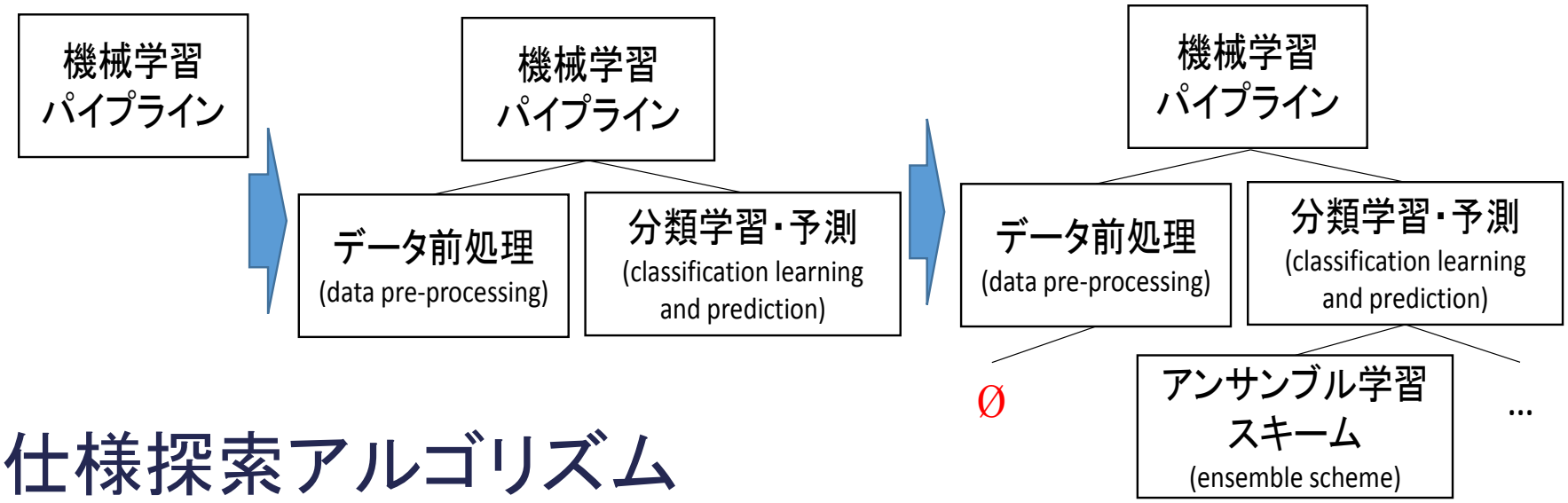
	TPOT	Auto-Weka	Auto-sklearn
機械学習ライブラリ	scikit-learn	Weka	scikit-learn
仕様探索空間の設定	木構造展開	木構造展開	木構造展開
仕様探索アルゴリズム	遺伝的アルゴリズム	SMAC	SMAC

タスクの展開とメソッドリポジトリに基づく仕様探索



メソッドリポジトリの記述に基づく機械学習パイプライン（一連の処理）の生成と実行

木構造展開に基づくパイプラインの生成



仕様探索アルゴリズム

- 遺伝的アルゴリズム（遺伝的プログラミング）→TPOTと同様
- 仕様変更の規則性（仕様書き換えルール）による仕様変更

仕様記述から実行可能コードへの変換

- 本手法では仕様記述は実行コードとは独立したものとする
  - 機械学習ライブラリに依存した仕様記述は他のライブラリで実行不可
  - ライブラリ自体の仕様変更で実行不可に
- 仕様記述から実行スクリプトへの変換例：

JSONによる決定木メソッドの仕様記述

```
{ "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},
   ...]
}
```

Python(scikit-learn)による実行コード

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score

training_data_x = loaded_training_set.x
training_data_y = loaded_training_set.y

clf = DecisionTreeClassifier(criterion='entropy',max_depth=3)
scores = cross_val_score(clf, X, y, cv=10)
```

おわりに

- 構成的メタ学習機構のための知識記述の検討
  - 入出力のデータやモデルを表す知識記述
  - 各処理（メソッド）の知識記述
  - 一連の処理の過程の知識記述
- 今後の課題
  - 知識記述のためのデータモデルに基づくタスクリポジトリ，メソッドリポジトリの作成
  - 仕様探索の実行機構，機械学習ライブラリへの翻訳機構の実装