

分散エッジ環境における機械学習実現最適化の検討 ～エッジ上で動作するアルゴリズム・オントロジーの決定と 転移学習適用による最適化検討～

森 郁海[†] 齊藤志保[†] 折本拓真[†] 伊藤岳広[†]

概要: 本稿は、分散エッジ環境における IoT システムの水平統合型共通プラットフォーム向けの機械学習アルゴリズムとオントロジーをリソース消費の観点から決定し、その妥当性を実機にて評価した結果をまとめたものである。評価の結果、機械学習アルゴリズムについては、訓練データ規模によらず学習データサイズが増加しにくく、行列計算等の高負荷な演算処理が原理的に発生しないものを選択するのが望ましいとした。オントロジーについては、エッジのリソースに応じて切り出し処理を行い、推論時には結果サイズを抑えることで、省リソース化と処理速度の低下の抑制を実現できる見込みを得た。また、訓練データ規模が小さい場合に機械学習の分類性能が低下する問題があるため、クラウド上で学習モデルを共有することで転移学習を自動的に実施可能な手法を提案した。

A Study on Optimisation of Machine Learning Realization In Edge Computing ~A Study on Optimisation by Deciding Edge executable Algorithm and Ontology and Applying Transfer Learning~

IKUMI MORI[†] SHIHO SAITO[†] TAKUMA ORIMOTO[†] TAKEHIRO ITO[†]

Abstract: We decided machine learning algorithms and ontology for the horizontal IoT platform in distributed edge environment from the view point of resource consumption. This decision was evaluated using Raspberry Pi 1 Model B as edge device. From the evaluation results, it is desirable to select machine learning algorithms that don't have the high processing load such as a matrix calculation. And it is required that the model data does not increase linearly with the training data size. As for the ontology, it is possible to realize both the resource saving and the high speed processing by excluding several nodes of ontology according to edge resources and suppressing the result size when inferencing. In addition, because there is a problem that prediction performance of machine learning deteriorates when the training data size is small, we propose a method that can automatically perform the transfer learning by sharing the model data on the cloud.

1. はじめに

2025 年の IoT(Internet of Things)における世界全体の経済効果は、FA(Factory Automation)が最大 444 兆円、次にスマートシティの 199 兆円、ウェアラブル(ヘルスケア)の 190 兆円と続き、非常に巨大なものになると予想されている[1].

IoT が発展するにつれ、機器の増加や高いリアルタイム性を要求するアプリケーションへの対応のために、エッジコンピューティングが必要である。さらに、ドメイン間の連携やデータの再利用を目的に、エッジコンピューティングを考慮した、IoT アプリケーションで共通的に使用する機能をまとめて提供する共通プラットフォーム[2]の標準化も進められている。この共通プラットフォームは、機械学習やオントロジーによる推論を使用して実現しており[3]、実行基盤として Microsoft IoT Edge[4]や AWS IoT Greengrass[5]等が存在する。

エッジコンピューティングで機械学習を利用する場合、クラウドで訓練を行い、エッジで予測や分類を行う形態が多いが、クラウドへデータを送信できないユースケースが存在するため、訓練処理もエッジで実施する形態も考慮す

る必要がある。従って、リソースの限られたエッジ上で機械学習やオントロジーによる推論を実行しようとする、以下の課題がある。

- リソース制約(演算能力、メモリ量、ストレージ量)
- 訓練処理をエッジで実行する場合、データの収集範囲が限られるため、訓練データが不足し、学習モデルの精度を高められない

本稿では、上記の課題に対して、「リソース制約を満たす機械学習アルゴリズムとオントロジーの決定方法」と「転移学習のための転移元学習モデルの特定方法」の2つを検討した内容を報告する。

2. 既存技術

クラウドとエッジが連携した IoT ソリューションの構築を支援するサービスとして、Microsoft IoT Edge や AWS IoT Greengrass が存在する。これらのサービスは、クラウドと機器をシームレスに接続し、クラウドの処理の一部を機器側へ拡張できる。Microsoft IoT Edge で機械学習を使用する場合、クラウド上で構築した学習モデルを機器にデプロイする形態を採る[6]。具体的には、機械学習はクラウドサー

[†] 三菱電機株式会社 情報技術総合研究所
Mitsubishi Electric Corporation Information Technology R&D Center

ビスである Azure Machine Learning[7]を使用して行い、訓練済みの学習モデルデータを Microsoft IoT Edge の機能を利用して機器へデプロイする。AWS IoT Greengrass の場合も同様に、クラウドサービスである Amazon SageMaker[8]などで学習モデルを構築し、機器へデプロイする[9]。この仕組みをオントロジーに対しても適用でき、クラウドでオントロジーの構築・管理を行い、エッジにオントロジーをデプロイすることができる。但し、デプロイ先のエッジのリソース制約を考慮し、適切な機械学習アルゴリズムとオントロジーを決定する必要がある。

また、上記の動作では、機械学習の訓練処理のために、訓練データをクラウドへ集積しなければならず、クラウドへのデータを送信できないユースケースには対応できない。そこで、訓練処理もエッジで実施する必要があるが、データの収集範囲に限られるため、訓練データの不足が懸念される。少数の訓練データでの学習手法として、次の3つのアプローチが考えられる。

- (1) 訓練データの増強(Data Augmentation)
- (2) 効率的に学習可能な訓練データの選択
(能動学習(Active Learning)[10]など)
- (3) 他のドメインで学習した学習モデルの使用
(転移学習[11])

(1)は、画像を入力とする場合に多く用いられ、元画像を反転、変形、ノイズ付与などして学習モデルのロバスト性を高める目的で利用する。エッジ上では訓練負荷そのものが問題となるため適用は難しい。(2)は、選択するという手法上、多くのデータが存在することが暗黙的に前提とされており、選択対象の母数が少ないと十分な効果を得られない。(3)は、Neural Network に関する研究が多いが、考え方は他の機械学習アルゴリズムにも適用可能である。しかし、転移元学習モデルをどのように特定するかが問題となる。

これらのことから、Microsoft IoT Edge や AWS IoT Greengrass のようなサービスで機械学習やオントロジーを扱う上では、「リソース制約を満たす機械学習アルゴリズムとオントロジーの決定方法」と「転移学習のための転移元学習モデルの特定方法」が課題となる。

本稿の3章、4章で、機械学習とオントロジーの各々の決定方法について述べる。5章で決定方法の妥当性に関する結論を述べ、6章で、転移学習のための転移元学習モデルの特定方法について述べる。

3. 機械学習アルゴリズムの決定方法の検討

3.1 機械学習アルゴリズムが持つべき機能

共通プラットフォーム向けの機械学習アルゴリズムが持つべき機能を挙げる。共通プラットフォームは、機器データに意味情報を付与するために機械学習(分類器)を使用する。この分類器は、例えばカメラの映像からの人物の行動識別や、音センサや赤外線センサの時系列データからの

ドアの開閉や人の出入りの特定といった複数のアプリケーションが共通的に使用するものである。分類には、回帰も使用可能である。回帰は、連続値の推定を対象とするが、推定結果を離散値(ラベル)に対応させることで分類器を構成できる。従って、共通プラットフォーム向けの機械学習アルゴリズムが持つべき機能は、分類および回帰である。但し、非線形分類問題のように、分類対象の統計的性質によっては、特定の機械学習アルゴリズムで精度低下が生じる恐れがある。この問題に対しては、IoT ソリューションの詳細な分析が必要なため、スコープ外とする。

3.2 リソース制約の判断軸と決定時の要件

エッジのリソース制約を定量的に把握するために、本稿では、以下の判断軸を使用する。

- 学習モデルサイズの増加率
- 訓練・分類処理時間
- 学習・分類時のメモリフットプリント

学習モデルサイズの増加率とは、訓練データ規模により学習モデルサイズがどの程度増加するかを表したものである。

共通プラットフォーム向けの機械学習アルゴリズムが満たすべき要件を以下のように設定する。

- 訓練データに比例して学習モデルサイズが増加しないこと
- 訓練は、行列演算などの高負荷処理が少なく、係数や重さを調整等軽量な処理であること
- 分類は、方程式への代入処理等の軽量な処理であること(但し、並列化が可能な場合はこの限りではない)
- 学習・分類時の基底のメモリフットプリントがエッジのメモリ量を超えないこと

これらの要件を満たすことで、上記評価軸の定量値が低く抑えられることを実機評価にて確認する。

3.3 機械学習アルゴリズムの定性的選定

IoT のデータ分析向けの機械学習アルゴリズム[12]から3.1 で述べた分類または回帰の機能を持つものを抽出し、3.2 の要件を基に、定性的な選定を行う。分類または回帰の機能を持つ機械学習アルゴリズムの一覧を表 1 に示す。表 1 の学習アルゴリズムの処理内容から導出される、リソース制約の判断軸の定性的性質を表 2 に示す。表 1、表 2 から 3.3 の要件を基に定性的に選定した結果を表 3 に示す。

表 1 機械学習アルゴリズムの機能

機械学習アルゴリズム	回帰	分類	並列化
k-Nearest Neighbors(k-NN)	×	○	○
Naive Bayes	×	○	×
Support Vector Machine(SVM)	×	○	×
Classification and Regression Trees(CART)	○	○	×
Random Forests	○	○	○
Bagging	○	○	○
Feed Forward Neural Network(FFNN)	○	○	○
Linear Regression	○	×	×
Support Vector Regression (SVR)	○	×	×

※ ○：可，×：不可

表 2 学習アルゴリズムの定性的性質

機械学習アルゴリズム	学習モデルサイズの増加率	訓練処理時間	分類処理時間
k-NN	高	短	やや長
	距離の計算に訓練データの座標情報が必要	訓練データの座標上へのプロットのみ	k の大きさに依存. 入力値付近に訓練データが密集すると, 距離の計算に時間を要する
Naive Bayes	低	短	短
	確率モデルのみを保持	分類対象の独立性を仮定し, 処理を単純化	確率モデルへの代入のみ
SVM	低	やや長	やや短
	超平面情報のみを保持するため	行列演算が発生. 多クラス分類時には, 手法によってはクラス数の組み合わせ分だけモデルを生成	2値分類は高速だが, 多クラス分類は低速なことがある (One-versus-one では, クラス数 K に対し最大 $K(K-1)/2$ 個の分類器を実行)
CART	中	やや短	中
	木のノード数や基準の種類に依存	木の構築は, 単純な演算処理 (遷移確率の設定)のみ	同左
Random Forests	中	やや短	中
	決定木の生成数, サンプル数に依存	決定木の生成数により変動するが, 木の構築は軽量	同左
Bagging	中	やや短	中
	用意する弱学習器の数に依存	訓練データをサンプリングし, 弱学習器へ投入するため, 弱学習器の訓練処理負荷に依存	同左
FFNN	中~高	長	中
	ノード数に依存. ノードごとにパラメータを保持	ノードごとにパラメータを学習するため, ノード数に比例し処理負荷が増加	通過する層の数に依存
Linear Regression	低	長	短
	説明変数分の係数, 切片情報のみ保持	行列演算が発生	方程式への代入のみ
SVR	低	やや長	やや短
	SVMと同様	SVMと同様	SVMと同様

3.4 評価

機械学習アルゴリズムの決定方法を, 3.2 の要件に加えて, 精度との関係を確認するために, 以下の項目について計測し, 評価する.

- 学習モデルサイズ
- 訓練・分類時間
- 訓練・分類時のメモリフットプリント
- 10-分割交差検証による平均正解率
- F 値(マイクロ平均)[a]

ハードウェア構成, ソフトウェア構成を表 4, 表 5 に示す. 本評価に使用するソフトウェアは, よく知られたものであるが, 評価結果がソフトウェアの実装に依存する部分があることに注意する.

- 評価パラメータの一覧を表 6 に示す. 機械学習アルゴリズムは, エッジ上での実行を考慮し, 線形モデルを基本とする.

a) 適合率と再現率の調和平均. 多クラス分類であるので, 平均値の算出にマイクロ平均を使用した。

表 3 機械学習アルゴリズムの定性的な選定

機械学習アルゴリズム	採否※
k-NN	×
	学習モデルサイズの増加率が高いため除外
Naive Bayes	×
	分類対象が独立性を持つことが少なく, 文書分類向けなので除外
SVM	○
	学習モデルサイズの増加率が低く, 分類処理時間が短いため選択
CART	×
	Random Forests に含まれるため除外
Random Forests	○
	訓練処理時間がやや短く, 学習モデルサイズの増加率と分類処理時間が中程度であり, 並列化が可能のため選択
Bagging	×
	Random Forests に含まれるため除外
FFNN	○
	総じてリソース消費が高い傾向にあるが, 並列化が可能のため選択
Linear Regression	○
	学習モデルサイズの増加率が低く, 分類処理時間が短いため選択
SVR	×
	SVM と同等のため除外

※ ○: 候補, ×: 除外

表 4 ハードウェア構成

エッジ	Raspberry Pi 1 Model B
CPU	700MHz シングルコア
主記憶	512MB
補助記憶	TS16GSDHC10(SDHC class 10 16GB)

表 5 ソフトウェア構成

		備考
OS	Raspbian	・ [13]より入手
ランタイム	Python 3	・ Raspbian にプリインストール済みのものを使用
ライブラリ	scikit-learn 0.19.2	・ 機械学習ライブラリ ・ [14]参照 ・ 依存ライブラリは割愛
機械学習用訓練・予測データセット	Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set	・ [15][16]参照 ・ 特徴空間: 561 次元 ・ 分類ラベル: 12 種類 ・ 訓練データセット: 7767 行 ・ 分類データセット: 3162 行

表 6 評価パラメータ

項目	値	備考
機械学習アルゴリズム	<ul style="list-style-type: none"> Random Forests Logistic Regression SVM Neural Network k-NN (参考値) 	<ul style="list-style-type: none"> 3.3 の選定結果を参照 Random Forests のツリー数は 10 Logistic Regression は線形回帰を使用 SVM は線形 SVC を使用し, 多クラス分類に one-vs-rest を選択 Neural Network の中間層は 3 層 (ノード数 100, 80, 80) k-NN は k=5 に設定 乱数シードを 0 に固定 上記以外のパラメータは, scikit-learn の既定値を使用[17]
訓練データ行数	<ul style="list-style-type: none"> 10, 100, 1,000, 5,000, 7,767 	<ul style="list-style-type: none"> 交差検証時は, 7,757 行すべてを使用

訓練データ行数と学習モデルサイズの変化を図 1 に示す。Linear Regression, SVM, Neural Network は、想定通り訓練データ行数によらず一定の値となった。Neural Network に関しては、中間層のノード数の影響から高めで推移している。Random Forests, k-NN についても想定通り訓練データ数に比例して増加することが確認できる。学習モデルサイズの増加率は、k-NN の方が高くこれも想定通りである。

訓練データ行数と訓練時間の変化を図 2 に示す。全てのアルゴリズムにおいて訓練データ行数に比例して増加した。K-NN, Random Forests, SVM, Linear Regression, Neural Network の順で訓練時間が長い傾向は想定通りである。

訓練データ行数と分類時間の変化を図 3 に示す。Linear Regression, SVM, Neural Network, Random Forests, k-NN の順で分類時間が長い傾向は想定通りである。k-NN も想定通り、訓練データ行数に比例して分類時間が増加する。

訓練データ行数と訓練時のメモリフットプリントの変化を図 4 に示す。k-NN は想定通り訓練データ行数に比例してメモリフットプリントが増加し、それ以外のアルゴリズムは多少の変動はあるものの、訓練データ行数によらず一定の値となった。Neural Network に関しては、中間層のノード数の影響から若干高めであった。

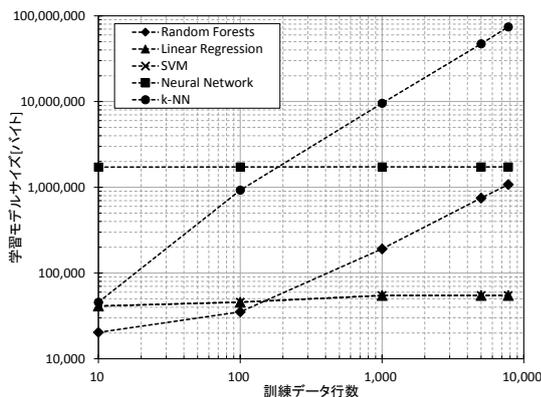


図 1 学習モデルサイズの変化

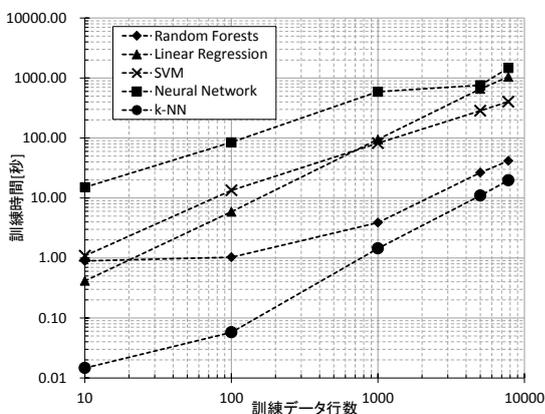


図 2 訓練時間の変化

訓練データ行数と分類時のメモリフットプリントの変化を図 5 に示す。大小関係は SVM ≒ Linear Regression < Neural Network ≒ Random Forests < k-NN となり、訓練時に学習するパラメータ数にほぼ比例すると考えられる。

機械学習アルゴリズムごとの 10-分割交差検証による平均正解率一覧を図 6 に示す。グラフ中のひげは、95%信頼区間である。Linear Regression, SVM, Neural Network は、Random Forests, k-NN と比較し 3~4% 正解率が高い。本評価では、訓練データ行数が最大でも 7,767 行と少数である。大規模な訓練データが取得可能な環境では、過学習に注意してアルゴリズムを選択すべきである。

訓練データ行数と F 値(マイクロ平均)の変化を図 7 に示す。いずれのアルゴリズムも訓練データ行数に比例して、性能が向上する傾向にあり、Linear Regression や SVM は、訓練データ行数が少数でも他のアルゴリズムに比べて高めの性能が出ている。Neural Network は、訓練データ行数が少ない場合に性能低下が発生しやすいことから、訓練データ行数がある程度ないと十分な性能を出すことができない可能性がある。

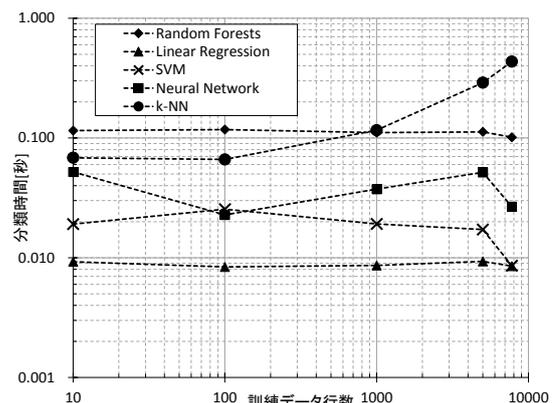


図 3 分類時間の変化

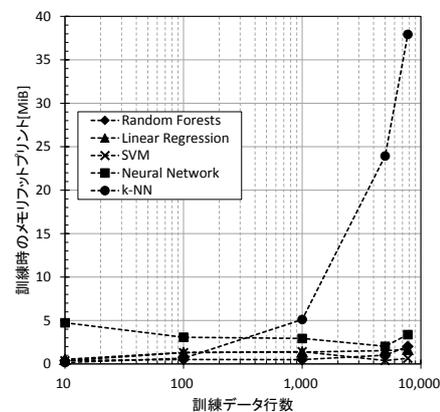


図 4 訓練時のメモリフットプリントの変化

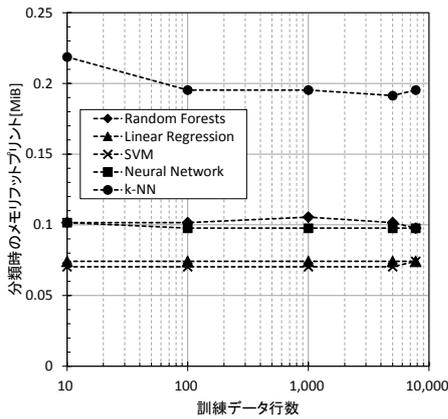


図 5 分類時のメモリフットプリントの変化

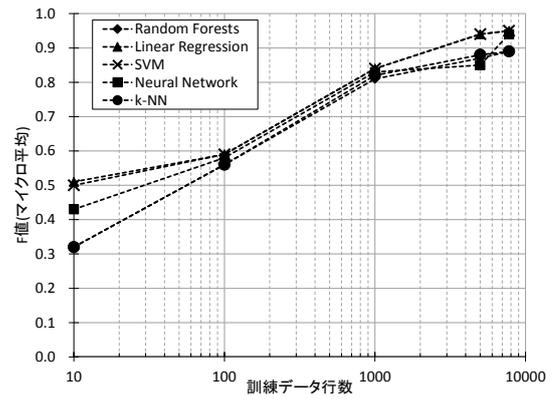


図 7 F 値の変化(マイクロ平均)

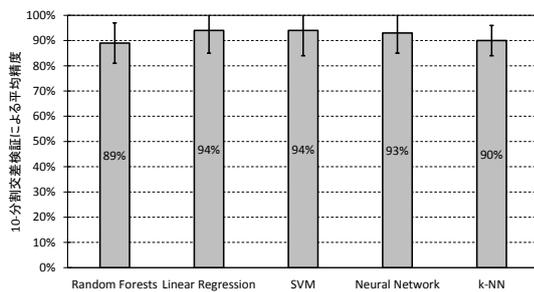


図 6 10-分割交差検証による平均正解率一覧

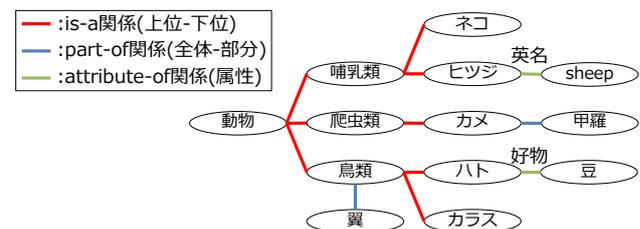


図 8 オントロジーの例

4. オントロジーの決定方法の検討

4.1 リソース制約の判断軸と要件

機械学習の場合と同様に、エッジのリソース制約を定量的に把握するために、本稿では、以下の判断軸を使用する。

- オントロジーデータサイズ
- オントロジー読み込み時間
- 推論時間

共通プラットフォーム向けのオントロジーが満たすべき要件を以下のように設定する。

- オントロジーサイズは、エッジの主記憶サイズ以下であること
- オントロジーは木構造が望ましく、かつ、木の深さが極端に深くなく、木の幅が極端に広くないこと
- 結果サイズが著しく増加する推論を想定しないこと

ここで、オントロジーは、ドメイン知識を概念(情報)間の関係[b]としてグラフ構造(図 8)で表現したものであるが、関係の性質上、木構造を持つことが多いと推察される。実機評価では、木構造を想定したオントロジーにおける要件の妥当性を確認する。

4.2 評価

オントロジーの決定方法の要件の妥当性を確認するために、以下の項目について計測し、評価する。

- オントロジーデータサイズ
- オントロジー読み込み時間
- 推論時間

ハードウェア構成は表 4 と同じである。ソフトウェア構成を表 7 に示す。OS、ランタイムは表 5 と同じである。評価パラメータの一覧を表 8 に示す。

オントロジーのノード数とオントロジーデータサイズ、読み込み時間、推論時間(全探索、一部探索)の変化を図 10 に示す。オントロジーデータサイズ、読み込み時間は、ノード数に比例して増加する。推論時間については、全探索の場合に著しく増える。一方、一部探索は木の深さが変わると変化するが増加割合は低い。即ち、推論における結果サイズが処理時間に支配的な影響を及ぼすと考えられる。

5. エッジ上で動作させる機能についての結論

機械学習およびオントロジーの要件は、実機評価の結果と整合が取れており、妥当であることを確認できた。

3.4 の実機評価の結果から、3.2 で示した機械学習アルゴリズムのリソース制約要件に照らして定量的に選択すると、評価に使用したソフトウェアとハードウェアの組み合わせにおいては、Linear Regression, SVM が有力であり、Neural Network, Random Forests はハイパーパラメータ次第では候補となる。

一方、4.2 の実機評価の結果から、オントロジーについては、4.1 で示したオントロジーのリソース制約要件に照らし

b) is-a 関係(上位-下位), part-of 関係(全体-部分), attribute-of 関係(属性)等

表 7 ソフトウェア構成

		備考
ライブラリ	rdflib 4.2.2	<ul style="list-style-type: none"> ・オントロジーを扱うライブラリ ・依存ライブラリは割愛

表 8 評価パラメータ

項目	値	備考
オントロジーのノード数	<ul style="list-style-type: none"> ・ 11 ・ 111 ・ 1,111 ・ 11,111 ・ 111,111 	<ul style="list-style-type: none"> ・ 各ノードのエッジ数は 10 ・ 木の深さは 1~5 ・ オントロジーの構造は, 図 9 を参照
推論	<ul style="list-style-type: none"> ・ 全探索 ・ 一部探索 	<ul style="list-style-type: none"> ・ 全探索は, オントロジーのノードすべてを返却する推論処理 ・ 一部探索は, 根を始点とし, 最深部の特定ノードへの探索を行う推論で, 経由したノードも返却する

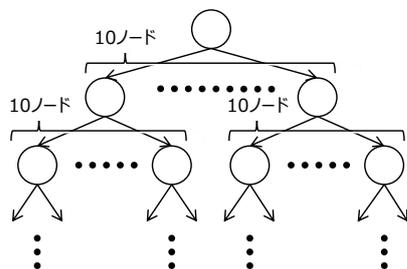


図 9 オントロジーの構成

合わせると, 数秒以内のリアルタイム性を求める場合は, 1000 ノード以下の木構造が望ましい. 但し, エッジが常時起動の場合は, 予めオントロジーをロードしておくことで, 処理時間の短縮が見込める. 推論に関しては, 結果サイズに比例して処理時間が増加するため, 推論時に探索範囲が必要以上に広がらないよう制御する必要がある. 結果サイズの増大が避けられない場合は, 結果の上位のみを返すか, 逐次返却[c]を実施すべきである.

特に, 探索範囲の抑制については, 推論精度との兼ね合いから以下の二つのアプローチが考えられる.

- (1) 関連性が低い途中結果を排除
- (2) 途中結果をシソーラス[d]により拡張

この 2 つのアプローチをバランスよく適用することで, 省リソース化と高精度な推論を両立が期待できる.

6. 転移学習の適用検討

2 章で述べたように, 転移学習を適用するにあたり, 転移に使用する学習モデルをどのように特定するかが課題となる. 本稿では, 次の手順で転移元の学習モデルを特定することを考える. まず, 転移先と同一のドメインの中から, ユーザーが要求する処理と同様な処理を行う訓練済み学習モデルを検索する. 発見したモデルが, ユーザーが指定し

c) 最初の n 件を最初に返却し, 残りを逐次返却

d) 単語の上位/下位, 部分/全体, 同義, 類義関係等で単語を分類し, 体系

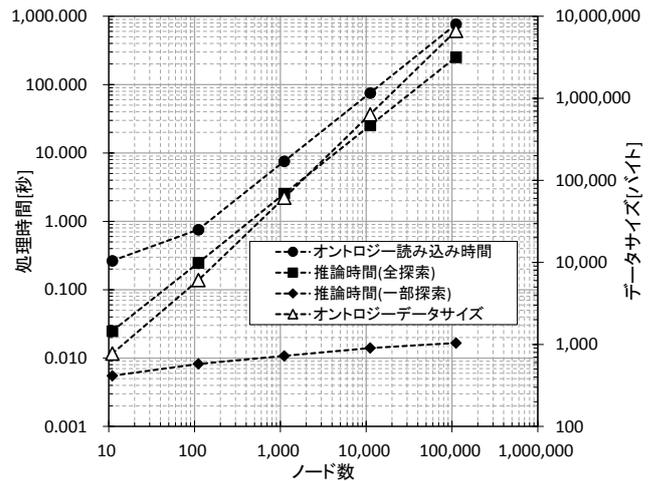


図 10 オントロジーノード数による特性

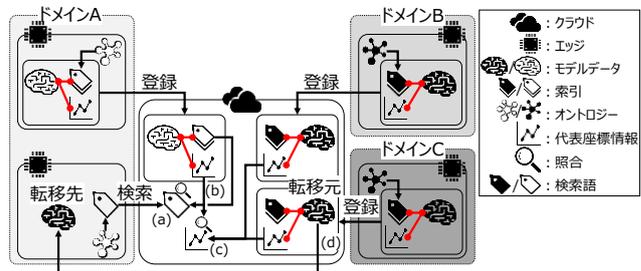


図 11 分散共通プラットフォームへの転移学習の適用例

た機械学習アルゴリズムと異なる場合, 発見した学習モデルを基準にして, 自または他のドメインの訓練済み学習モデルの中から, 以下の条件を満たすものを検索する.

- 分類結果の数が同じ
- 各分類結果(出力)に対応する入力データの領域の形状が似ている学習モデル

検索結果のうちユーザーが指定した機械学習アルゴリズムのものを転移先学習モデルの候補とする.

図 11 は, これらの処理を模式図で表したものである. 処理の詳細手順を以下に示す.

■ 転移元エッジによる学習モデルの登録

- (1) 予め作成・共有済みのオントロジーを使用して, 自身が保有する学習モデルに関する索引を生成
- (2) 後述の訓練データの代表座標情報を計算
- (3) 索引, 代表座標情報, 学習モデルをクラウドへ登録

■ 転移先エッジによる学習モデルの検索要求

- (1) オントロジーを使用して, 自身が要求する学習モデルに関する検索語を生成
- (2) 検索語と要求する学習モデルの種類(機械学習アルゴリズム)等をクラウドへ送信
- (3) クラウドから学習モデルを得る

づけた辞書

■ クラウドによる学習モデルの検索

- (1) 登録済みの索引から、学習モデルを検索(図 11(a))
- (2) 要求する学習モデルの種類と一致すれば、発見した学習モデルをエッジへ返却し、終了
- (3) 要求する学習モデルの種類と一致しない場合、発見した学習モデルに紐づく代表座標情報を得る(図 11(b))
- (4) 取得した代表座標情報と最も類似する代表座標情報を持つ学習モデルを特定(図 11(c))
- (5) 要求する学習モデルの種類と一致すれば、特定した学習モデルを返却(図 11(d))

但し、上記処理は、以下が成立していることを前提とする。

- (1) 転移先と同一のドメインに対して、オントロジーを使って検索して見つかる訓練済み学習モデルは、ユーザが要求する学習モデルに近いものである
- (2) 分類結果に対応する入力データの領域の形状が似ていることを代表座標情報で判断できる

(1)に関して、オントロジーは、シソーラスのようなアプリケーションの特徴を表すキーワードとそれから連想される語をまとめたものを想定する。例えば、アプリケーションを屋内の人の行動検知やビルの入退出管理とすると、それらのキーワードは、屋内、人、行動検知、ビル、入退出などとなる。これらのキーワードをシソーラスで拡張し、索引とすることで、検索時にヒットしやすくなる。

(2)に関しては、線形分離可能な問題を対象とし、訓練データの分布が真の入力データの領域に均一に分布していると仮定すると、代表座標情報は入力データの領域の特徴を保存できる。本稿では、議論の対象外とするが、線形分離不可能な問題に対しては、代表座標情報ではなく、SVMの分離超平面のような各分類結果領域の境界の形状で判断できると考えられる。

代表座標情報の計算方法について説明する。n次元の特徴空間上で、最大m個の分類クラスを持つとき、分類クラスiにおける訓練データjの座標 C_j^i の定義を数式1に示す。

$$\{C_j^i = (jx_1, \dots, jx_n) \in \mathbb{R}^n \mid 1 \leq i \leq m, 1 \leq j \leq N_i, i \in \mathbb{N}, j \in \mathbb{N}\} \dots (数式 1)$$

ここで、 N_i は、分類クラスiに分類される訓練データ数である。

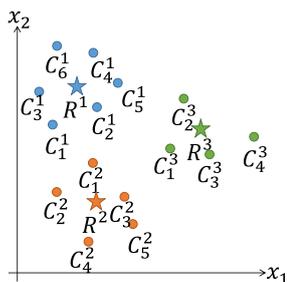


図 12 訓練データと代表座標の関係

分類クラスiにおける代表座標は、例えば、分類クラスiの訓練データの重心を使用することができる。代表座標 R^i の定義を数式2に示す(図 12)。

$$\{R^i = (x_1, \dots, x_n) \in \mathbb{R}^n \mid x_k = \frac{\sum_{j=1}^{N_i} jx_k}{N_i}, 1 \leq k \leq n, k \in \mathbb{N}\} \dots (数式 2)$$

訓練データは、ドメインによって定義域や値域が異なることが予想されるため、代表座標の位置関係が同じでも、単純に比較ができない。そこで、代表座標を、例えば区間[0,1]に収めるような正規化処理を施す。全てのクラスについて代表座標を正規化し(R_{norm}^i とする)、まとめたものを代表座標情報Fとする(数式3)

$$F = \{R_{norm}^i \mid 1 \leq i \leq m, i \in \mathbb{N}\} \dots (数式 3)$$

次に代表座標情報同士の類似度の算出方法を考える。類似度は、(高次)相関係数を用いる手法もあるが、直感的に理解するために、幾何的に捉える方法を考える。代表座標情報は、代表座標を頂点とする(高次元)図形として表現することができる。この図形同士が「似ている」場合、類似度が高いとみなす。しかし、代表座標情報の座標軸に対応する特徴量は、ドメインごとに異なる。従って、平行・回転移動、拡大・縮小・反転変形を行い比較する必要がある(図 13)。ここでは簡単のため、上記の移動・変形が発生しない場合を考える。2つの代表座標情報 F_1, F_2 の図形の頂点座標(代表座標)が誤差を許容していくつ重なるかで類似度を測る。 F_1 のある代表座標 R_{norm}^p と F_2 のある代表座標 R_{norm}^q のユークリッド距離が最も近くなる(p,q)の組み合わせを見つけ、その距離が閾値 T_1 以下であれば(数式4)、点数を与える(図 14)。点数は、距離に近いほど高くなるよう重み α をつける。

$$\|R_{norm}^p - R_{norm}^q\| \leq T_1 \Rightarrow \alpha$$

但し、 $(\alpha \propto^{-1} \|R_{norm}^p - R_{norm}^q\|) \dots (数式 4)$

全ての代表座標に対して点数を計算し、その合計値が閾値 T_2 以上で、かつ、最も合計点数が高い F_x を持つ学習モデルを転移元とする(数式5)。

$$\exists F_x = \max_{\forall i} \{Score(F_i)\} \geq T_2 \Rightarrow F_x \dots (数式 5)$$

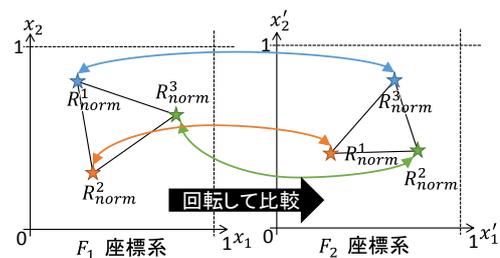


図 13 回転移動による比較

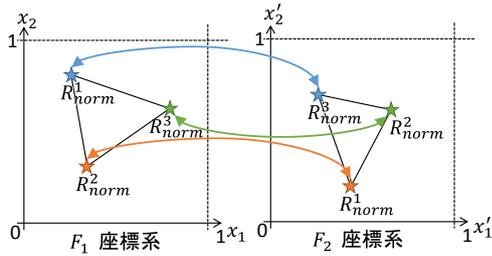


図 14 組み合わせの発見と類似度の比較

7. まとめ

本稿では、分散共通プラットフォーム向けの機械学習アルゴリズムとオントロジーの決定方法を検討し、その妥当性を実機評価した。評価の結果、機械学習アルゴリズムは、訓練データ規模によらず学習データサイズが増加しにくく、行列計算等の高負荷な演算処理が発生しないものを選択するのが望ましいとした。オントロジーについては、エッジのリソースに応じて切り出し処理を行い、推論時には結果サイズを抑えることで、省リソース化と処理速度の低下の抑制を実現できる見込みを得た。また、訓練データ規模が小さい場合に機械学習の分類性能が低下する問題があるため、クラウド上で学習モデルを共有することで転移学習を自動的に実施可能な手法を提案した。

今後は、転移学習の適用手法の追加検討とセキュリティを考慮した方式の検討を実施する。

参考文献

- [1] 総務省：日本における情報通信分野の現状と課題,総務省(オンライン),入手先
(<http://www.tele.soumu.go.jp/resource/j/equ/mra/pdf/29/j/03.pdf>) (参照 2018-11-30).
- [2] 総務省：情報通信審議会 情報通信技術分科会 技術戦略委員会 (第 12 回) (文書審議) 第 2 次中間報告書 (案) 参考資料, 総務省(オンライン), 入手先
(http://www.soumu.go.jp/main_content/000439135.pdf) (参照 2018-11-30).
- [3] 稲田修一：M2M/IoT 教科書, インプレス(2015).
- [4] Microsoft Azure：Azure IoT Edge とは, Microsoft (オンライン),入手先 (<https://docs.microsoft.com/ja-jp/azure/iot-edge/about-iot-edge>) (参照 2018-12-14)
- [5] Amazon Web Services：AWS IoT Greengrass とは, Amazon Web Services (オンライン),入手先
(https://docs.aws.amazon.com/ja_jp/greengrass/latest/developerguide/what-is-gg.html) (参照 2018-12-14)
- [6] Microsoft Azure：チュートリアル: Azure Machine Learning を IoT Edge モジュールとして展開する (プレビュー), Microsoft (オンライン),入手先 (<https://docs.microsoft.com/ja-jp/azure/iot-edge/tutorial-deploy-machine-learning>) (参照 2018-12-14)
- [7] Microsoft Azure：Azure Machine Learning サービスの概要, Microsoft (オンライン),入手先 (<https://docs.microsoft.com/ja-jp/azure/machine-learning/service/overview-what-is-azure-ml>) (参照 2018-12-14)
- [8] Amazon Web Services：Amazon SageMaker とは, Amazon Web Services (オンライン),入手先

- (https://docs.aws.amazon.com/ja_jp/sagemaker/latest/dg/whatis.html) (参照 2018-12-14)
- [9] Amazon Web Services：チュートリアル: Azure Machine Learning を IoT Edge モジュールとして展開する (プレビュー), Amazon Web Services (オンライン),入手先
(https://docs.aws.amazon.com/ja_jp/greengrass/latest/developerguide/ml-inference.html) (参照 2018-12-14).
- [10] 中村篤祥：特集『能動学習』,情報処理, Vol. 38, No. 7, pp. 557-588 (1997).
- [11] 神嶋 敏弘：転移学習, 人工知能学会誌, vol.25, no.4, pp.572-580 (2010).
- [12] Mahdavinejad, M.S., Rezvan, M., Barekatain, M., et al.: Machine learning for Internet of Things data analysis: A survey, Digital Communications and Networks, DOI: 10.1016/j.dcan.2017.10.002(2017).
- [13] The Raspberry Pi Foundation：NOOBS - New Out Of the Box Software, The Raspberry Pi Foundation (オンライン),入手先
(https://downloads.raspberrypi.org/NOOBS_latest) (参照 2018-09-14).
- [14] Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: Machine Learning in Python, The Journal of Machine Learning Research, Vol.12, pp. 2825-2830(2011).
- [15] Reyes-Ortiz, J-L., Oneto, L., SamÀ, A., et al.: Transition-Aware Human Activity Recognition Using Smartphones, Neurocomputing, Volume 171, pp. 754-767(2016).
- [16] UC Irvine Machine Learning Repository: Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set, UC Irvine Machine Learning Repository(online), available from (<http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions>) (accessed 2018-11-30).
- [17] scikit-learn developers: API Reference, scikit-learn(online), available from (<https://scikit-learn.org/0.19/modules/classes.html>) (accessed 2018-11-30).