

AI/IoT ソリューションへの転移学習適用における 学習モデル検索システムの提案と評価

森郁海^{1,†1,a)} 伊藤岳広^{1,b)} 中村嘉隆^{†1,c)} 稲村浩^{†1,d)}

本稿は、IoT 機器上で AI を利用するソリューションにおいて、転移学習を成功させるために、入出力データ分布の類似性に着目して転移候補となる学習モデルを検索するシステムを提案し、評価したものである。入出力データ分布の類似性の評価手法を 4 つ提案し、転移先/元の差異を吸収するための入出力の変換精度と、転移のための所要時間を評価した。評価の結果、転移元候補の検索空間が大きく、探索に即時性が必要なユースケースには、探索時間が転移先/元データ数に対し変動しにくい、確率密度推定を用いる 2 つの方式が適しているとの結論を得た。

1. はじめに

AI/IoT 市場の拡大に伴い、IoT 機器上で AI を利用するソリューションが増加している¹⁾。特に、一般物体認識(一般画像認識)と呼ばれる、制約のない実世界の画像から物体の一般的な名称や状態を説明する技術が注目されており、自動運転や製造・生産管理等への応用が期待されている。一方で、学習に必要な大量のデータの取得が困難な領域があり、学習データ不足を解消する研究が進められている。

一般物体認識では、畳み込みニューラルネットワーク(CNN)を使用することが多い²⁾。CNN では、画像の局所的な特徴抽出を担う畳み込み層と、局所ごとに特徴をまとめるプーリング層を繰り返すことで特徴量を取り出し、全結合層と出力層で特徴量に基づく分類を行う。また、CNN は、高い汎用性を持つため、学習済ネットワーク(Pre-trained network)を流用(転移)することができる³⁾。

Pre-trained network をより活用するために、転移先のデータセットを用いて更にネットワークの学習を進める Fine-tuning のアプローチが広く用いられる。Fine-tuning では、Pre-trained network の出力層を転移先のものに付け替え、その他の部分は学習済みのパラメータを初期値として用い、誤差逆伝搬法による学習を進める。但し、CNN の学習は初期値依存性が強いいため、特に訓練データが少ない場合はできるだけよい初期値を得ることが重要となる。即ち、Pre-trained network の作成に用いられた入力の中に、転移先のタスクが含まれている必要がある⁴⁾。

本稿は、上記のような転移学習を成功させるために、転移元と転移先タスクの入力の類似性を判定し、適切な学習済みモデルを発見する検索システムを提案・評価することを目的とする。一方、IoT 機器では、処理負荷の観点からサポートベクターマシン(SVM)や k 近傍法などの古典的な機械学習アルゴリズムを用いることもある⁵⁾。本稿では、CNN の他、古典的な機械学習アルゴリズムの中で初期値依存性

が高いものも対象とする。

2. 関連研究

2.1 転移学習の分類と定義

転移学習は、転移先/元の入力のラベルの有無によって、4 つの分類が存在する(表 1)。ここで、S,T は転移元、転移先を、ラベルは教示情報を表す。ラベルは、主にクラス分類における識別に使用する⁶⁾。4 分類の中で最も研究例が多いのは、S+T+である(表 1 太枠)。また、文献 2)によれば、転移先/元ドメインの違いを入力データ分布の違いと定義しているが、入力データの定義域の違いを考慮した研究は少ない。例えば、カラー画像を入力とすると、その定義域は RGB 要素それぞれ 8bit(256 段階)であり、画像内の識別対象によらず定義域は同じである。この時、特徴量にヒストグラムを用いる場合、識別対象によって RGB 要素に対する画素値の出現分布が異なるため、データ分布の違いが生じる。このように、暗黙的に似たドメイン間の転移を想定している場合が多い。

しかし、EMS(エネルギーマネジメントシステム)におけるエネルギー需要予測や、製造装置の故障診断、IoT 家電の自動制御のような IoT アプリケーションは、入力に加速度や赤外線といった画像以外のデータを扱うことも多い。さらに、転移元候補が入力によって制限されるのは好ましくない。本稿では、研究例が多い S+T+を対象とし、定義域の違いも考慮する。但し、学習モデルの転移までを対象とし、Fine-tuning についてはスコープ外とする。

表 1 転移学習の分類

		転移先のラベル	
		あり (+)	なし (-)
転移元のラベル	あり (+)	S+T+	S+T-
	なし (-)	S-T+	S-T-

1 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, 5-1-1, Ofuna, Kamakura City, 247-8501, Japan.
†1 公立はこだて未来大学大学院
Future University Hakodate Graduate School, 116-2, Kamedanakano-cho, Hakodate, Hokkaido, 041-8655, Japan.

a) Mori.Ikumi@ab.MitsubishiElectric.co.jp
b) Ito.Takehiro@da.MitsubishiElectric.co.jp
c) y-nakamr@fun.ac.jp
d) inamura@fun.ac.jp
e) 入力にラベルが与えられるか否かで教師あり(なし)学習と呼ぶ

2.2 Frustratingly Easy Domain Adaptation

文献 7)は、訓練データを変換し、一般的な学習器を使用して転移学習を行う非常に簡潔な手法を提案している。まず、訓練データを次のように変換する。

● 転移先： $(\vec{x}^{(T)}, y^{(T)}) \rightarrow ((\vec{x}^{(T)}, \vec{0}, \vec{x}^{(T)}), y^{(T)})$

● 転移元： $(\vec{x}^{(S)}, y^{(S)}) \rightarrow ((\vec{x}^{(S)}, \vec{x}^{(S)}, \vec{0}), y^{(S)})$

上付きの(T), (S)は転移先、転移元であることを表し、 \vec{x} は、出力ラベルyに対応する入力ベクトルである。 $\langle \rangle$ はベクトルの連結であり、「共通」、「転移元ドメイン」、「転移先ドメイン」に対応した三つの部分で構成される元の3倍の長さのベクトルである。この変換したベクトルを入力として、通常の学習器で訓練する。すると、転移先/元で共通する特徴は、ベクトルの連結の「共通」部分を使用して学習される。一方、転移先/元で働きが異なる特徴は、転移先/元のどちらか一方の値が0となる「転移元ドメイン」または、「転移先ドメイン」の部分から自然に利用される。

前述の Pre-trained network と同様に、「共通」部分を持つ転移元を発見済であることを前提としている。もし、任意の転移元に対して網羅的に学習モデルを作成する場合、作成したすべての学習モデルに対し識別精度の確認を行うことになり、非効率である。

2.3 Pre-trained network

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)⁸⁾は、2010年から始まった大規模画像認識の競技会である。2012年以降、ILSVRC では AlexNet, GoogLeNet, VGGNet, ResNet などの CNN ベースの Pre-trained network を用いる方法が主流であり、フルスクラッチよりも Pre-trained network と Fine-tuning を組み合わせた場合のほうが非常に良い精度を発揮している²⁾。

Pre-trained network は、出力層の1~2つ手前の全結合層を経験的に用いることが多い。但し、ImageNetでの学習のため、ImageNetがカバーしない範囲の分類は期待した精度が出ない可能性がある。従って、転移先のタスクがImageNetに含まれるかを確認する必要がある。

2.4 非定常環境下での学習

転移学習では、転移先/元ドメインの違いをデータ分布の違いと定義するが、これを定量的に分析する手法が存在する。文献 9)では、多くの場合、訓練データとテストデータ(推論時に使用するデータ)が同じ確率分布に従わず、教師付き学習法の汎化性能が低下すると述べている。確率分布が異なる状況とは、入力変数の確率分布が変化する共変量シフトと、クラス分類問題においてクラス事前確率が変化するクラスバランス変化の2つである。

本稿における共変量シフト、クラスバランス変化は、転移学習時の転移先/元の訓練データ(入力ベクトルとラベルの組)数の差異に相当する。一般に、転移学習時は転移元の訓練データ数が多く、転移先は少ない。転移先タスクが転移元に含まれるかを確認するために、前述の状況下で訓練

データがラベルごとに同じ確率分布に従う(真の確率分布が一致または類似することを確認する必要がある)。

共変量シフト、クラスバランス変化は、訓練時と推定時の入力確率密度関数をカーネル密度推定などで外挿・推定し、カルバック・ライブラー距離やピアソン距離、 L^2 距離などの距離指標を使用して検知することができる。

本稿では、この方法を高次元の入力データに対しても適用できるようにする。

3. 提案手法

2章の議論より、転移学習を効率的に成功させるためには、Fine-tuningの前に転移先タスクが転移元に含まれていることを判断する必要がある。転移先タスクの含有判断は、2.4節で述べた方法以外にも、いくつか考えられる。処理速度や精度の観点からそれらの評価を行い、それぞれの方法の特性を把握する。

本稿で提案する学習モデル検索システムは、転移元の訓練データと転移先の少数の観測データを使用してデータ分布の類似性を自動的に判定し(図 1(a)), 転移先タスクの含有判断を行う。この処理により、転移元の検索空間を絞り込み、転移が成功することを示す正の転移の判定負荷を軽減する(図 1(b))。

また、学習モデル検索システムは、入出力の定義域の違いを吸収するために、基底変換や正規化、転移先/元間のラベルの対応関係の特定などを行う。以下に処理手順を示す(図 2中の番号と対応)。

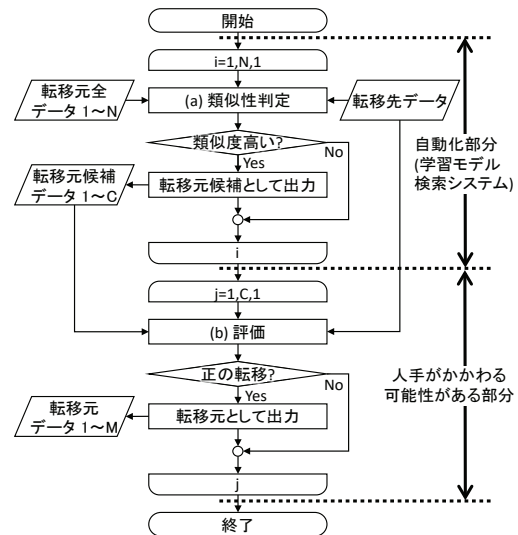


図 1 学習モデル検索システムの位置付け

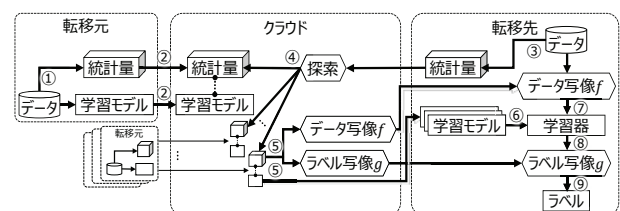


図 2 学習モデル検索システム

- 処理①： 転移元は、訓練データから学習モデルと類似性の比較に必要な統計量を作成
- 処理②： 統計量と学習モデルをクラウドへ送信
- 処理③： 転移先は、観測データから類似性の比較に必要な統計量を作成し、クラウドへ送信
- 処理④： クラウドは、転移先/元の統計量を用いて類似性を比較し、転移元の候補となるものを探索
- 処理⑤： 転移元候補に対するデータ写像 f (転移先から転移元への入力変換)、ラベル写像 g (転移元から転移先への出力変換)を作成
- 処理⑥： 転移元の学習モデルを入力とし、学習器を構築
- 処理⑦： 転移先の観測データをデータ写像で変換後、学習器に入力
- 処理⑧： 学習器の出力をラベル写像で変換
- 処理⑨： 最終ラベル出力

3.1~3.2で処理①~処理③で実施する基底変換、正規化、特徴量の算出と比較方法を説明する。3.3で、処理④の類似度による転移元候補の探索方法を説明する。3.4で、手順処理⑤のデータ写像 f 、ラベル写像 g について説明する。3.5で処理⑦~処理⑨の推論処理を説明する。

3.1 基底変換、正規化

ラベルごとに特徴量ベクトル(訓練データからラベルを除いたデータの分布を比較するために、基底変換により座標系をそろえる。例えば、主成分分析(「PCA」と表記)などを用いて、特徴量ベクトルの各要素の分散値(固有値)が大きい順で、正規直交基底を得る。具体的には、 p 次元ユークリッド空間 \mathbb{R}^p 上の特徴量ベクトル \vec{x} を、 m 次元主成分空間 \mathbb{Z}^m 上のベクトル \vec{z} へ変換する(図3)。

ここで、 \vec{z} の第 i 主成分を z_i 、 z_i の寄与率を PV_i 、累積寄与率を CPV_m と表記する。この変換により、主成分間同士が無相関化される。 \vec{z} の次元数を m とすると、 $1 \leq m \leq p$ 、 $0 < CPV_m \leq 1$ を満たす。特に、 $m < p$ の場合、次元削減と呼ぶ。PCAにより、転移先/元の特徴量ベクトル空間軸が寄与率の降順でソートされる。

次に、特徴量ベクトルの分布のスケールにとらわれない比較をするために、特徴量ベクトルを正規化する。 z_i を $z_{\min} \leq z_i \leq z_{\max}$ となるよう、数式1のように正規化する(図4)。 \hat{z} を正規化したものを \vec{z} と表記する。

$$\hat{z}_i = C(z_i, z_{\min}, z_{\max})$$

$$s.t. C(x, C_{\min}, C_{\max}) = \frac{x - \min(x)}{\max(x) - \min(x)} (C_{\max} - C_{\min}) + C_{\min} \dots \text{数式 1}$$

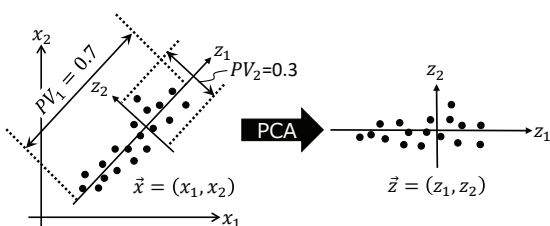


図3 基底変換の例

3.2 統計量の算出

本稿では、次の4方式の統計量の算出手法を述べる。

- 方式(1)： ベクトルのグレースケール化と相関値比較
- 方式(2)： 独立特徴量の確率密度推定量の相関値比較
- 方式(3)： 独立特徴量の統計的仮説検定による比較
- 方式(4)： 平均ベクトルのコサイン類似度による比較

方式(1)は、特徴量ベクトルを寄与率で重み付けしたグレースケール画像に変換し、その分布の推定量で相関値比較する。特徴量ベクトルを2次元空間に射影するので、人の目で類似性を確認できる。但し、低次元への射影時に情報が損失するため、特徴の違いがセンシティブである(寄与率が小さく拮抗している)場合は、識別できない可能性がある。

方式(2)、方式(3)は、基底変換により特徴軸を互いに無相関化したことを利用して、特徴量ベクトルの要素ごとに類似性を判定し、各々の結果を寄与率による重み付けをして最終決定する方式である。方式(2)は関連研究の考え方に基づき、類似性の判定にデータ分布の推定量を用いるため、転移先/元でデータ数が異なる場合にも対応できる。方式(3)は厳密な類似性の判定が必要な場合を想定し、統計的仮説検定を使用する。検定手法によっては、サンプルの信頼性が高く、ノイズが少ないことを要求することに注意する。

方式(4)は、高次元空間上のデータの類似度の測定によく使用されるコサイン類似度を用いるものである。本稿では、ラベルごとの特徴量ベクトルの平均値をとり、コサイン類似度を算出する。非常に処理が軽量であるが、ベクトルのなす角が支配的となるため、平均ベクトルが同方向であるようなデータ分布は、識別が困難となる。

3.2.1 方式(1)の統計量

方式(1)では、ラベル y に対する特徴量ベクトル \vec{z}_y ごとにグレースケール化を行い、そのカーネル密度推定量を統計量とする。次の手順に沿ってグレースケール化を行う。

手順(1)： $z_{\min} = 0$ 、 $z_{\max} = 255$ で正規化した \vec{z} の天井関数 $\lceil \vec{z} \rceil$ をとり8bitに量子化

手順(2)： PV で重み付けしたグレースケール画像に変換

グレースケール画像は、ユニットと呼ぶ小さな領域の集合からなる。 i 行 j 列のユニット $U(i, j)$ の画素値は $\lceil z_j \rceil$ 、高さは1、幅 w_j は数式2となる(図5)。

$$w_j = \begin{cases} \lceil PV_j \times 100 + 0.5 \rceil & , w_j > 0 \\ 1 & , w_j \leq 0 \dots \dots \dots \text{数式 2} \end{cases}$$

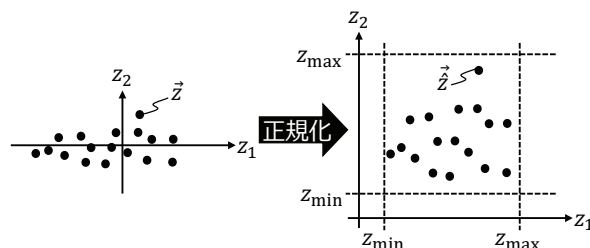


図4 正規化の例

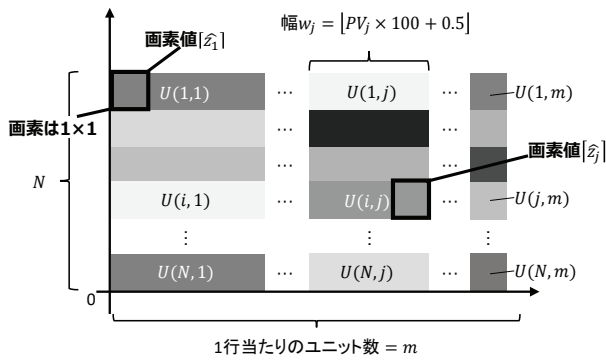


図 5 寄与率による重み付き 8bit 量子化

以降、グレースケール画像の*i*行*j*列の画素値を $g_{i,j} \in \mathbb{G} (1 \leq i \leq N, 1 \leq j \leq \sum_{j=1}^m w_j, N$ はラベルごとの特徴量ベクトル数)と表記する。

次に、転移先/元の \mathbb{G} の類似度を比較するために、ヒストグラムを求める。ただし、特徴量ベクトルから生成されたヒストグラムは、本来の母集団の特徴を反映していない可能性があるため、母集団の確率密度関数の推定を行う。 \mathbb{G} を母集団の標本としてカーネル密度推定量 $\hat{f}_h(x)$ を数式 3 で定義する。

$$\hat{f}_h(x) = \frac{1}{|\mathbb{G}|h} \sum_{g_{i,j} \in \mathbb{G}} K\left(\frac{x-g_{i,j}}{h}\right) \dots \dots \dots \text{数式 3}$$

$|\mathbb{G}|$ は画素数、 h はバンド幅(平滑化パラメータ)、 K はカーネル関数である。本稿では、 K にガウス関数を用いる。類似度の比較のために必要な統計量は、この $\hat{f}_h(x)$ となる。

3.2.2 方式(2)の統計量

方式(2)では、転移先/元の \vec{z}_y の要素 z_i ごとに確率密度推定量を算出し、統計量とする。まず、 $z_{\min} = 0, z_{\max} = 1$ で正規化した z_i に対し、カーネル密度推定量 $\hat{f}_h(x)$ を算出し、確率密度関数を推定する(数式 4)。全ての*i*に対し $\hat{f}_h(x)$ を計算し、これを統計量とする。

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{x \in z_i} K\left(\frac{x-x_i}{h}\right) \dots \dots \dots \text{数式 4}$$

3.2.3 方式(3)の統計量

方式(3)では、転移先/元の \vec{z}_y の要素 z_i ごとに統計的仮説検定を行う。従って、 $z_{\min} = 0, z_{\max} = 1$ での正規化のみを行い、統計量を算出せず、 \vec{z} そのものをクラウドへ送信する。但し、統計的仮説検定における検定精度の低下を抑制するために、外れ値やノイズの除去、データの補間や外挿を行ってもよい。

3.2.4 方式(4)の統計量

方式(4)では、転移先/元の \vec{z}_y の平均ベクトルを統計量とする。 $z_{\min} = 0, z_{\max} = 1$ で正規化した \vec{z} に対して、相加平均ベクトル $\bar{\vec{z}}$ を数式 5 で算出し、統計量とする。

$$\bar{\vec{z}} = \frac{\sum \vec{z}}{N} \dots \dots \dots \text{数式 5}$$

3.3 類似度の算出と転移元候補の探索

3.3.1 方式(1)の類似度と転移元候補の探索

転移先/元の $\hat{f}_h^{(T)}(x), \hat{f}_h^{(S)}(x)$ を使用してピアソン相関係数による類似度比較を行う。文献 9)では、類似度の評価にカルバック・ライブラー距離やピアソン距離、 L^2 距離を用いる方法が述べられている。しかし、IoTでの転移を考える場合、転移元のデータ数が転移先に比べ少ない状況($N^{(T)} < N^{(S)}$)が多く想定される。このため、各画素値に対する出現頻度分布に差異が生じ、前述の距離では、類似性を正しく判断できない。そこで、2 データ間の増減の関係に着目し、ピアソン相関係数を使用する。

まず、ピアソン無相関検定で2群間に相関があるかを検定する。検定の結果、無相関ではないと判定した場合、相関係数を類似度とする(数式 6)。但し、ラベル y_k に対応する $\hat{f}_h^{(T)}(x) \rightarrow \hat{f}_h^{(T)}(x)_{y_k}$ 、ラベル y_l に対応する $\hat{f}_h^{(S)}(x) \rightarrow \hat{f}_h^{(S)}(x)_{y_l}$ と表記している。一方、無相関であるとは言えない(帰無仮説を棄却できない)場合、類似度を0と定義する。無相関検定および相関係数の計算に用いる標本は、ヒストグラムのbin幅で十分であるので、 $x = 1, \dots, 255$ を代入したときの $\hat{f}_h^{(T)}(x), \hat{f}_h^{(S)}(x)$ の値を使用する。

$$\text{score}_{(y_k^{(T)}, y_l^{(S)})} = \text{pearsonr}(\hat{f}_h^{(T)}(x)_{y_k}, \hat{f}_h^{(S)}(x)_{y_l}) \dots \text{数式 6}$$

$y_k^{(T)}$ に対応する $y_l^{(S)}$ は、 $\text{score}(y_k^{(T)}, y_l^{(S)})$ が最も高いものとする。具体的には、転移先の探索始点を変更しながら($y_k^{(T)}$ を $k = 1, 2, \dots$ と順に変化させる)、類似度が最大のラベル $y_l^{(S)}$ を順次特定していく(図 6)。

この時、転移先のあるラベルに対応する入力と最も類似するものから転移元のラベルを決定する場合(図 6 の {used}で制御)、本来対応すべきラベルが他のラベルとの対応関係に使用され、類似度計算精度が低下する恐れがある。そこで、転移先のラベルの探索順序を決めておき、始点をずらしながら、転移元のラベルに対し総当たりで類似度を計算していく。転移先のラベルの探索順序を $(y_r^{(T)}, y_{1+r}^{(T)}, \dots, y_{(q^{(T)}-1+r) \bmod q^{(T)}}^{(T)})$ としたとき、 r を $0, 1, \dots, q^{(T)} - 1$ とずらし、転移先/元のラベルの対応関係の中でラベルごとの類似度の和が最大のものを類似度比較に用いる(図 6①)。ここで、 $q^{(T)}$ は、ラベルの種類数である。この例では、 $\{y_0^{(T)}, \dots, y_{q^{(T)}-1}^{(T)}\}$ の $q^{(T)}$ 種類のラベルが存在する。

このようにして、転移先と全ての転移元との類似度を算出し、転移が有効な転移元候補を絞り込んでいく。例えば、類似度の上位*M*件($1 \leq M$)や、閾値*T*以上の類似度を持つものを候補とする。

3.3.2 方式(2)の類似度と転移元候補の探索

z_i の寄与率 PV_i で重みを付けたピアソン相関係数を計算し、類似度とする(数式 7)。この時、ピアソン無相関検定および相関係数の計算に用いる標本は、 $x = 0, 0.001, \dots, 1$ を代入したときの $\hat{f}_h^{(T)}(x), \hat{f}_h^{(S)}(x)$ の値を使用する。

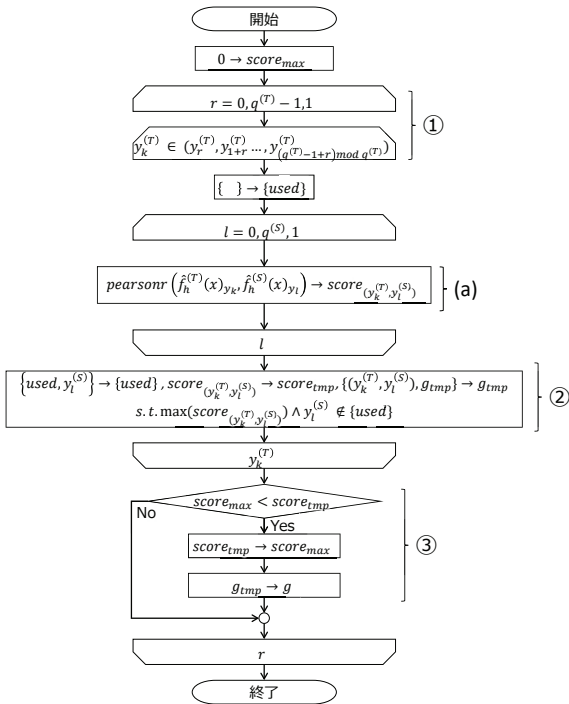


図 6 類似度の算出と比較(転移元候補一つ当たり)

$$\text{score}_{(y_k^{(T)}, y_l^{(S)})} = \sum_{i=1}^{\min(m^{(T)}, m^{(S)})} PV_i^{(T)} \times \text{pearsonr}(\hat{f}_h^{(T)}(x)_{y_k}, \hat{f}_h^{(S)}(x)_{y_l}) \dots \text{数式 7}$$

そして、図 6(a)を図 7 のように変更する。

3.3.3 方式(3)の類似度と転移元候補の探索

$\hat{z}_i^{(T)}$, $\hat{z}_i^{(S)}$ を標本として、統計的仮説検定によって類似度を算出する。統計的仮説検定では、帰無仮説 H_0 、対立仮説 H_1 を定め、 H_0 を棄却することで H_1 を採択する。検定の結果を類似度算出のため、 H_0 が棄却された場合を0、棄却できない場合を1と定義し2値化する。但し、検定結果が1であっても H_0 を採択しないことに注意する。検定の標本は、 $(\hat{z}_i^{(T)})_{y_k}, (\hat{z}_i^{(S)})_{y_l}$ を用いる。下付きの y_k や y_l は、ラベル y_k, y_l に対応する特徴量ベクトル \hat{z} の要素 z_i である。3.3.2と同様に、検定結果に寄与率で重み付けして類似度を算出する(数式8)。Testは検定の結果を2値化したものである(数式9)。

$$\text{score}_{(y_k^{(T)}, y_l^{(S)})} = \sum_{i=1}^{\min(m^{(T)}, m^{(S)})} \{PV_i^{(T)} \cdot \text{Test}((\hat{z}_i^{(T)})_{y_k}, (\hat{z}_i^{(S)})_{y_l})\} \dots \text{数式 8}$$

$$\text{Test} = \begin{cases} 1, & H_0 \text{を棄却できない場合} \\ 0, & H_0 \text{を棄却する場合} \end{cases} \dots \text{数式 9}$$

そして、図 6(a)を図 8 のように変更する。

検定方法を選定する際、転移先/元ドメインの特性によって、以下のような条件を考慮する必要がある。

- (1) 正規性を仮定できない
- (2) サンプル数が異なる(独立2標本, 対応のない標本)

表 2は、主な検定方法の一覧である。本稿では、(1)(2)の条件を仮定し、対応のないノンパラメトリック検定を使用する(表 2太枠内)。マンホイットニのU検定(以降「U検定」と表記)では、 H_0 を「両標本が同じ母集団から抽出された」、 H_1 を「両標本が異なる母集団から抽出された」とする。2標本コルモゴロフ-スミルノフ検定(以降「KS検定」と表

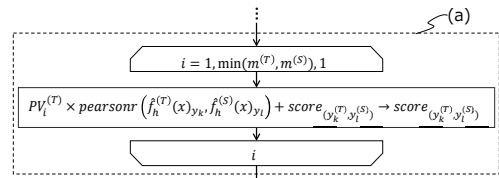


図 7 類似度の比較処理の変更

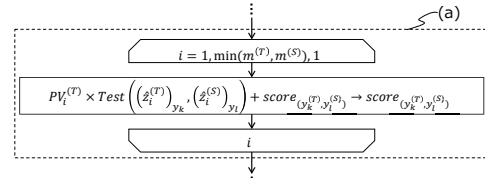


図 8 類似度の比較処理の変更

表 2 検定方法一覧

		検定法(例)	
パラメトリック検定	2群の平均値の差の検定	対応あり	対応のあるt検定
		対応なし	ウェルチのt検定
ノンパラメトリック検定	2群の代表値の差の検定	対応あり	ウィルコクソンの符号順位和検定 符号検定
		対応なし	マンホイットニのU検定 2標本コルモゴロフ-スミルノフ(KS)検定

記)では、 H_0 を「両標本の母集団の確率分布が等しい」、 H_1 を「両標本の母集団の確率分布が等しくない」とする。

3.3.4 方式(4)の類似度と転移元候補の探索

$\hat{z}^{(T)}$, $\hat{z}^{(S)}$ のコサイン類似度を、類似度とする(数式10)。

$$\text{score}_{(y_k^{(T)}, y_l^{(S)})} = \cos \left(\left(\frac{\hat{z}^{(T)}}{\|\hat{z}^{(T)}\|} \right)_{y_k}, \left(\frac{\hat{z}^{(S)}}{\|\hat{z}^{(S)}\|} \right)_{y_l} \right) = \frac{\sum_{i=1}^{\min(m^{(T)}, m^{(S)})} \{(\hat{z}_i^{(T)})_{y_k}, (\hat{z}_i^{(S)})_{y_l}\}}{\sqrt{\sum_{i=1}^{\min(m^{(T)}, m^{(S)})} \{(\hat{z}_i^{(T)})_{y_k}\}^2} \cdot \sqrt{\sum_{i=1}^{\min(m^{(T)}, m^{(S)})} \{(\hat{z}_i^{(S)})_{y_l}\}^2}} \dots \text{数式 10}$$

そして、図 6(a)の式を数式 10 に変更する。

3.4 データ写像f, ラベル写像gの作成

3.4.1 データ写像fの作成

データ写像fの生成に必要な統計量は、転移元と転移先の \vec{x} , \vec{z} の軸の対応関係と、正規化前の \vec{x} の定義域の最大値($\max(x_i)$)と最小値($\min(x_i)$)である(図9)。 \vec{x} , \vec{z} の軸の対応関係は、転移先の元の座標系→転移先の基底変換後の座標系→転移元の基底変換後の座標系→転移元の元の座標系の順で特定していく。図9の例の場合、各軸の分散の大小関係を見ることで、数式11のように特定する。

$$(x_2^{(T)}, x_1^{(T)}) \leftrightarrow (z_1^{(T)}, z_2^{(T)}) \leftrightarrow (z_1^{(S)}, z_2^{(S)}) \leftrightarrow (x_1^{(S)}, x_2^{(S)}) \Rightarrow (x_2^{(T)}, x_1^{(T)}) \leftrightarrow (x_1^{(S)}, x_2^{(S)}) \dots \text{数式 11}$$

この対応関係を $\mathcal{R}(i) = j$ と表すと、図9の例の場合、 $\mathcal{R}(2) = 1, \mathcal{R}(1) = 2$ となる。但し、iは転移先の軸の添え字($x_1^{(T)}$ であれば1)、jは、転移元の軸の添え字($x_2^{(S)}$ であれば

2)である。また、 $\max(x_i)$ 、 $\min(x_i)$ は、特徴量ベクトルの各要素に対して算出する。

\mathcal{R} 、 $\max(x_i)$ 、 $\min(x_i)$ は、学習モデル検索システムの処理②において各方式の統計量と共にクラウドへ送信する。

クラウドでは、データ写像 f を $\max(x_i^{(S)})$ 、 $\min(x_i^{(S)})$ 、および、 \mathcal{R} を使用し、数式12のように生成する。

$$f = \begin{cases} \mathcal{D}(x_i^{(T)}) = \mathcal{C}(x_i^{(T)}, \min(x_i^{(S)}), \max(x_i^{(S)})) \\ \mathcal{R}(i) = j \end{cases}$$

$$(x_1^{(T)}, \dots, x_p^{(T)}) \rightarrow (\mathcal{D}(x_{\mathcal{R}(1)}^{(T)}), \dots, \mathcal{D}(x_{\mathcal{R}(p)}^{(T)})) \dots \text{数式 12}$$

$p^{(T)}$ は、転移先の特徴量ベクトル \vec{x} の次元数である。 \mathcal{C} は、数式1の定義と同じである。

3.4.2 ラベル写像 g の作成

ラベル写像 g は、転移元候補について、最も類似度が高いラベルの対応関係から生成する(図6②③)。具体的には、 $y_r^{(T)}$ から順に対応するラベル $y_l^{(S)}$ を特定していき、対応ラベルの組の集合 $\{(y_k^{(T)}, y_l^{(S)}) | 0 \leq k \leq q^{(T)} - 1, 0 \leq l \leq q^{(S)} - 1\}$ として作成する。

3.5 推論処理

転移先では、転移した f 、 g および学習モデルを使用して、推論処理を行う。まず、転移先でデータを取得すると、 f を使用して、転移元の入力形式に変換する。例えば、データ分布が図9の場合、 f は数式13のようになる。この f を使用し、 $x_1^{(T)}$ 軸を $x_2^{(T)}$ 軸へ、 $x_2^{(T)}$ 軸を $x_1^{(T)}$ 軸へと入れ替え、正規化を行う。

$$(x_1^{(T)}, x_2^{(T)}) \rightarrow (\mathcal{D}(x_2^{(T)}), \mathcal{D}(x_1^{(T)}))$$

s. t. $\mathcal{R}(1) = 2, \mathcal{R}(2) = 1 \dots \dots \dots$ 数式 13

次に、変換した入力データを学習器に入力し、推論を行う。最後に、推論して得られた出力ラベルを、 g のラベルの対応関係の集合 $\{(y_k^{(T)}, y_l^{(S)})\}$ から、転移先のラベルを特定する。例えば、 $g = \{(\text{りんご}, \text{自動車}), (\text{みかん}, \text{バイク}), (\text{ばなな}, \text{自転車})\}$ だとすると、推論でラベル「みかん」が出力された場合、ラベル「バイク」を出力する。

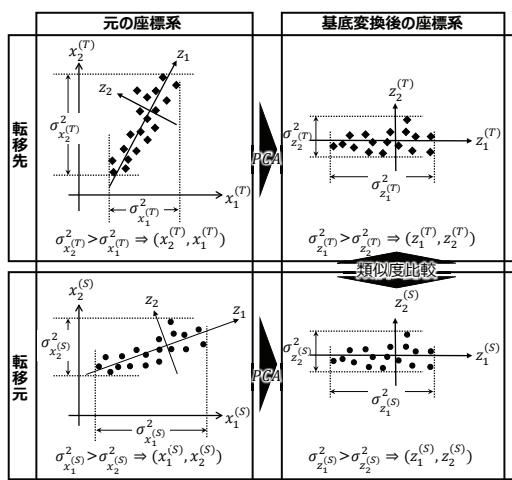


図9 基底変換前後の軸の対応関係

4. 評価

4.1 評価項目

本稿では、以下の2点についての評価を行う。

- (1) 線形分離不可能なデータ分布での転移元の特定
- (2) 実データセットでの転移元の特定

(1)では、代表値だけでは特定できない場合が存在する⁶⁾ため、同心円状に広がるデータ分布の類似性を判定できるかを確認する。ノイズを加えたテスト用データセットを用いて、データ数を変化させながら、性能変化を見る。

(2)では、実環境上での効果を確認するために、オープンデータセットから転移先/元の訓練データをラベルごとにランダムサンプリングして評価する。この時、ラベル、特徴軸をランダムに入れ替え、データ写像 f 、ラベル写像 g を正しく生成できるかを確認する。

4.2 評価指標

正しく転移元を特定できるかを確認するために、以下の2点の評価指標を用いて性能を計測する。

- (1) ラベル、特徴軸の対応関係の正答率
- (2) 転移元1つあたりの類似度の算出所要時間

(1)では、異なるドメイン間での転移を想定し、入力データ分布から生成したデータ写像 f 、ラベル写像 g が正しく生成されているかを評価する。

(2)では、処理のボトルネックを特定する。

4.3 評価環境

評価に使用したPCのハードウェア環境、ソフトウェア環境、評価用データの一覧を表3、表4、表5に示す。

4.4 評価パラメータ

4.1で述べた評価項目に対するパラメータを表6、表7に示す。

4.5 評価結果

評価(1)のラベルの対応関係の正答率、所要時間を表8、図10に示す。特徴軸の対応関係の正答率については、各軸の分散がほぼ等しく、どちらに対応しても問題ないため計測しない。

評価(2)のラベル、特徴軸の対応関係の正答率、所要時間を図11、図12、図13に示す。特徴軸の対応関係の正答率は、累積寄与率90%までの軸を対象に正誤を評価した。

表3 ハードウェア環境

CPU	Intel Core i5-3570 (3.4GHz, 4コア)
メモリ	32GB
ストレージ	Samsung SSD 850 Pro 1TB

表4 ソフトウェア環境

		備考
OS	Windows 7 Professional	・64bit版を使用
実装言語	Python3.6	
主な使用ライブラリ	scipy ¹¹⁾	・統計処理に使用
	pillow ¹²⁾	・グレースケール化に使用
	scikit-learn ¹³⁾	・主成分分析に使用

5. 考察

5.1 評価(1)

ラベルの対応関係の正答率をみると、方式(3)[U検定]と方式(4)で正答率が低いことがわかる。U検定は、順位和に基づくため、平均値が同じで分散が異なる(値の相対的な順位が同じ)分布が識別できず、誤りが増加したと考えられる。方式(4)も同様に、平均ベクトル同士が同じ方向であるとき、識別が困難となる。

所要時間をみると、方式(1)が際立って長い。これは、グレースケール化に伴う処理負荷が高いためである。また、方式(2)で事前処理の割合が高いが、これは確率密度推定の処理不可が高いことに起因する。但し、推定後の検定に用いるサンプル数は固定値となるため、訓練データ数によらず探索時間は一定に抑えられる。一方、方式(3)は、事前に検定できないため、探索時間が訓練データ数に依存する。方式(4)は、平均ベクトル・コサイン類似度算出処理負荷が他の方式に比べ低いため、ほとんど時間が変化しない。

5.2 評価(2)

ラベルの対応関係の正答率をみると、転移先の訓練データ数に比例して正答率が高くなる傾向にある。中でも方式(2)が早い段階で100%となる。これは、確率密度推定によ

るノイズ除去とデータ外挿による効果である。方式(1)は、2次元射影後に推定処理を1度しか行わないので、訓練データ数が少ない状況下で正答率が低く、データ外挿による効果が得られにくい。方式(3)は正答率のばらつきが大きく、訓練データ数を増やしても正答率が上がりにくい。検定に用いる訓練データに外れ値が含まれ、検定力が低下したと考えられる。方式(4)も正答率が安定しない。訓練データ数が増えると平均ベクトルが真の値と近くなるが、ノイズの影響も受けるためである。

特徴軸の対応関係の正答率をみると、累積寄与率65%前後(5軸)までは確実に対応付けがなされている。それ以降の正答率が低いのは、分散が拮抗しているためだと考えられる。但し、現実的には、累積寄与率65%前後の特徴軸の対応ができれば、学習モデル構築における支配的な特徴量との対応がとれており Fine-tuning で補正可能と判断できる。

所要時間をみると、方式(3)の探索時間が評価(1)に比べ著しく増加しているのがわかる。評価(1)での考察通り、訓練データ数が増えたためである。その他の方式については、評価(1)の考察通りである。

表 5 評価データ

	データセット	備考
評価(1)	scikit-learn の make_circles ¹⁴⁾	・転移先: noise=0.02, factor=0.2 ・転移元: noise=0.1, factor=0.2
	Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set ¹⁵⁾¹⁶⁾	・特徴空間: 561次元 ・分類ラベル: 12種類 ・データ数: $N^{(D)} = \sum_{i=1.0}^{12.0} N_i = 7767$ $N_{1.0}=1226$ $N_{5.0}=1423$ $N_{9.0}=75$ $N_{2.0}=1073$ $N_{6.0}=1413$ $N_{10.0}=60$ $N_{3.0}=987$ $N_{7.0}=47$ $N_{11.0}=90$ $N_{4.0}=1293$ $N_{8.0}=23$ $N_{12.0}=57$

表 6 評価(1)のパラメータ

	転移先	転移元
訓練データ数	$N^{(T)} = 10, 20, 30, 40, 50$	$N^{(S)} = 100$
特徴次元	・ $p^{(T)} = 2$ ・ $m^{(T)} = 2$ ・ $CPV_m^{(T)} \geq 0.9$	・ $p^{(S)} = 2$ ・ $m^{(S)} = 2$ ・ $CPV_m^{(S)} \geq 0.9$
	転移元との特徴軸の対応関係を入れ替える	
ラベル	{0.0, 1.0}	{0.0, 1.0}
	転移元とのラベルの対応関係を入れ替える	

表 7 評価(2)のパラメータ

	転移先	転移元
訓練データ数	$N^{(T)} = 120, 240, 353, 463, 570$ $N^{(T)} = \sum_{i=1.0}^{12.0} \min(N_i, C)$, $C=10, 20, 30, 40, 50$	$N^{(S)} = 952$ $N^{(S)} = \sum_{i=1.0}^{12.0} \min(N_i, 100)$
特徴次元	・ $p^{(T)} = 561$ ・ $m^{(T)} = 30, 37, 39, 40, 42$ ・ $CPV_m^{(T)} \geq 0.9$	・ $p^{(S)} = 561$ ・ $m^{(S)} = 41$ ・ $CPV_m^{(S)} \geq 0.9$
	転移元との特徴軸の対応関係をランダムに入れ替える	
ラベル	{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0}	{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0}
	転移元とのラベルの対応関係をランダムに入れ替える	

表 8 評価(1)のラベルの対応関係の正答率

	$N^{(T)}=10$	$N^{(T)}=20$	$N^{(T)}=30$	$N^{(T)}=40$	$N^{(T)}=50$
方式(1)	○	○	○	○	○
方式(2)	○	○	○	○	○
方式(3)[U検定]	×	×	×	×	×
方式(3)[KS検定]	×	○	○	○	○
方式(4)	○	×	○	×	○

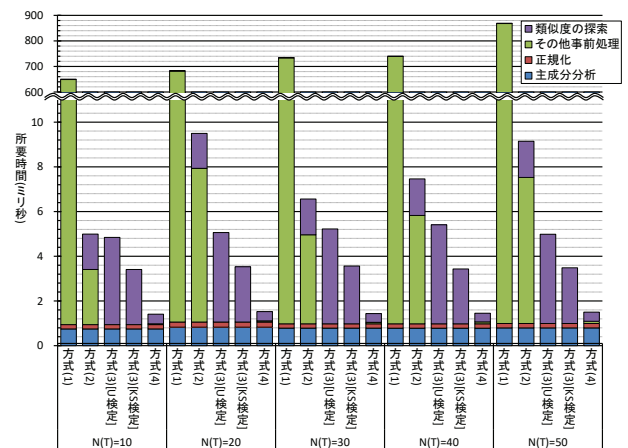


図 10 評価(1)の所要時間

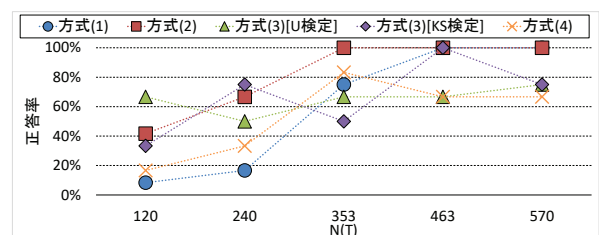


図 11 評価(2)のラベルの対応関係の正答率

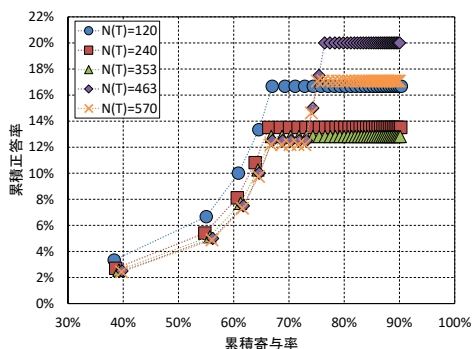


図 12 評価(2)の特徴軸の対応関係の累積正答率

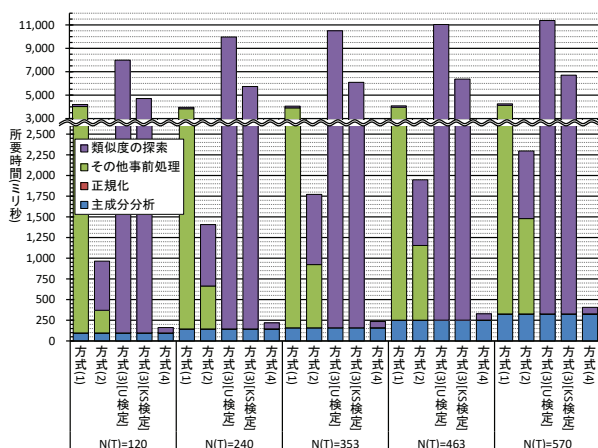


図 13 評価(2)での所要時間

5.3 方式ごとの特性とIoTへの適用について

本評価環境においては、3.2で述べた方式の特性を概ね確認できた。方式(1)は、評価(2)のような寄与率に差が少ない特徴軸を含み、転移先データ数が少ない場合に性能を出しにくい。類似度の探索時間は一定であり特徴次元数に依存しない。方式(3)は、評価(1)(2)のようにノイズを多く含むサンプルに弱い。方式(4)は高速であるが、評価(1)のように線形分離不可能なデータ分布の識別が困難となる。方式(2)は、その他の方式に比べ良い安定的に性能を出すことができるが、特徴次元数に比例して探索時間が増加する。

但し、転移元の探索時間は、転移元1つあたりにかかる類似度算出時間(評価における類似度の探索時間)と転移元候補数の積とほぼ等しい。このことから、転移元候補の探索空間が大きく、探索に即時性が必要なユースケースには、方式(1)や方式(2)が向いていると考えられる。

6. まとめ

本稿では、AI/IoTソリューションへの転移学習適用を目的とした学習モデル検索システムを提案・評価した。転移先/元の入出力データ分布の類似性の評価・探索手法を4つ提案し、ドメイン差異を吸収するために必要な入出力の変換精度と、類似度の探索所要時間を評価した。評価の結果、

転移元候補の探索空間が大きく、探索に即時性が必要なユースケースには、探索時間が転移先/元データ数に対し変動しにくい、確率密度推定を用いる2つの方式が適しているとの結論を得た。

今後は、転移元特定精度のさらなる向上を目指し、データのノイズに対しロバスト性を持たせる改良に取り組む。

参考文献

- 1) 総務省：平成30年版情報通信白書 第2章 ICTによる新たなエコノミーの形成, 総務省(オンライン), 入手先
(<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/pdf/n2100000.pdf>) (参照 2019-07-12).
- 2) 中山英樹：深層畳み込みニューラルネットワークによる画像特徴抽出と転移学習, 電子情報通信学会技術研究報告, vol.115, no.146, pp.55-59(2015).
- 3) 神鷹 敏弘：転移学習, 人工知能学会誌, vol.25, no.4, pp.572-580(2010).
- 4) Donahue, J. Jia, Y. Vinyals, O., et al.: DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, In Proc. ICML'14, vol.32, pp. I-647-I-655(2014).
- 5) Zhou, B. Lapedriza, A. Xiao, J., et al.: Learning Deep Features for Scene Recognition using Places Database, In Proc. NIPS'14, vol.1, pp. 487-495(2014).
- 6) 森郁海, 齊藤志保, 折本拓真, 伊藤岳広：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~, 情報処理学会研究報告, Vol. 2019-DPS-177, No. 2, pp. 1-8 (2019).
- 7) Daum'e, H., III: Frustratingly Easy Domain Adaptation, in Proc. of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 256-263 (2007).
- 8) ImageNet, ImageNet Large Scale Visual Recognition Challenge (online), available from (<http://www.image-net.org/challenges/LSVRC/>) (accessed 2019-09-17).
- 9) 杉山 将, 山田 誠, ドゥ・プレシ マーティヌス・クリストフェル, リウ ソン：非正常環境下での学習：共変量シフト適応, クラスバランス変化適応, 変化検知, 日本統計学会誌, vol.44, no.1, pp.113-136(2014).
- 10) 株式会社データサイエンス研究所：検定の種類と選択方法, 株式会社データサイエンス研究所(オンライン), 入手先
(<https://www.statweb.jp/method/sentakuhouhou>) (参照 2019-05-27)
- 11) SciPy.org, SciPy developers (online), available from
(<https://www.scipy.org/>) (accessed 2019-06-21).
- 12) Pillow(PIL Fork), Alex Clark and Contributors (online), available from (<https://pillow.readthedocs.io/en/latest/index.html>) (accessed 2019-06-21).
- 13) 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).
- 14) sklearn.datasets.make_circles, scikit-learn developers(online), available from (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_circles.html) (accessed 2019-06-27).
- 15) Reyes-Ortiz, J-L., Oneto, L., Sam'Á, 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).