事例研究［論文］

複数ドメインからの転移を想定した回帰手法とタクシーの需要予測
—Multi Source Two-stage TrAdaBoost.R2の提案—

柴田　頼仁，岩本　大輝，小森　晴菜，多賀　友哉，鈴木　秀男

1. はじめに

近年，カーシェアサービスなどの普及に伴い移動手段が多様化し，移動手段間でのユーザー獲得競争は激化しつつある．タクシー業界では的確な需要予測に基づいた効率的なタクシー配置がユーザー獲得のための鍵となると考えられている [1]．また，配車支援サービス [2] や需要予測サービス [3] などの実社会ですでに導入されている需要予測サービスもいくつか存在しており，需要予測のニーズの高さがうかがえる．

学術界に目を向けると，Yao et al. [4] の研究で，深層学習をベースにした精度の高いタクシー需要予測モデルが提案されている．しかし，この論文では，祝日や雨天時などといったデータが少なく，需要パターンが規則的でない特殊な条件下での需要予測が困難であることが課題として言及されている．

そこで本研究では，特殊な条件下での需要予測に焦点を当て，転移学習を用いた回帰アルゴリズムを提案する．特殊な条件下での需要予測は，学習データとテストデータの性質が異なるため，通常の機械学習的アプローチをとることができない．そのため，学習データとテストデータの性質が完全に一致することを仮定しない転移学習的アプローチを採用する．本研究ではAdaBoost.R2 [5] を転移学習に応用した回帰手法である Two-stage TrAdaBoost.R2 [6] をベースに，複数ドメインからの転移を可能にした Multi Source Two-stage TrAdaBoost.R2 を提案する．この提案手法は，転移元であるソースドメインが複数個存在するケースを想定しており，すべてのソースドメインを区別せずに扱う先行研究 [6] の手法に比べ，より柔軟な転移を可能にしている．

本研究では，経営科学系研究部会連合協議会主催，令和元年度データ解析コンペティションで提供されたタクシーのプローブデータを使用し，元日の明治神宮・浅草寺周辺のタクシー需要予測を行った．元日は公共交通機関が終日運行していること，神社などに多くの初詣客が参拝しに来ることから，上記の地域では通常時とは異なる性質をもったタクシー需要が発生することが見込まれる．2016 年 12 月と 2017 年元日，2017 年 12 月のデータを使用し，2018 年元日の 1 時間ごとのタクシー需要量を予測したところ，提案手法は先行研究の手法や転移学習を使わない手法を上回る予測精度を得ることができた．

2. 先行研究

1 節で述べたように，タクシーの需要予測は大規模なデータを活用して高精度な予測を実現している．一方，Yao et al. [4] や Horne and Manzenreiter [7] で言及されているように祝日や雨天時，大規模イベント開催時などのデータが少なく，特殊な条件下での需要予測は精度が低下してしまうという課題がある．

転移学習に関する研究は多岐にわたっているが，本研究で提案する手法はドメイン適応と呼ばれる分野に位置づけられる．ドメイン適応とは，予測対象であるターゲットドメインとは異なる分布であるソースドメインの情報を活用して，モデルの予測性能を向上させることを目的とする分野である [8]．さらにドメイン適応の中でも想定する状況は多岐に及んでいる．本手法は，少量のターゲットドメインデータと大量のソースドメインデータが訓練データに存在している状況を想定し

しばた　らいじん，こもり　はるな，たが　ゆうや
慶應義塾大学大学院理工学研究科
〒 223–8522　神奈川県横浜市港北区日吉 3–14–1
いわもと　ひろき
早稲田大学創造理工学部
〒 169–8555　東京都新宿区西早稲田 3–4–1
すずき　ひでお
慶應義塾大学理工学部
〒 223–8522　神奈川県横浜市港北区日吉 3–14–1
受付 20.7.25　採択 20.11.5

た教師あり転移学習手法である．この状況を想定した手法の一つに TrAdaBoost がある [9]．TrAdaBoost は AdaBoost を転移学習のために改良した分類手法である．また，似た手法として，AdaBoost.R2 [5]（アルゴリズムは付録参照）を転移学習のために改良した回帰手法である Two-stage TrAdaBoost.R2 がある [6]．Two-stage TrAdaBoost.R2 のモデル推定法をアルゴリズム 1 に示す．Two-stage TrAdaBoost.R2 が AdaBoost.R2 と異なる点は次のとおりである．

- モデルの構造が 2 段階になっており，ソースドメインとターゲットドメインの重みの比率を変動させた AdaBoost.R2 の学習器を複数個作成する
- AdaBoost.R2 の学習時，ソースドメインの重みは変化しない
- 予測時は，クロスバリデーションによって求めたターゲットドメインの予測誤差が最も小さくなった AdaBoost.R2 の学習器を用いて予測値を算出する

モデルの構造が 2 段階になっているのは TrAdaBoost にもない Two-stage TrAdaBoost.R2 独自のものである．これは，AdaBoost.R2 や TrAdaBoost のようにターゲットドメインと同時にソースドメインの重みを更新すると，ソースドメインの重みが小さくなりすぎてうまく学習できないという課題を解決するための工夫である．

このような工夫により，Two-stage TrAdaBoost.R2 は教師あり転移学習における回帰問題に対応した手法として提案された．しかし，この手法は一つのソースドメインからの転移のみを仮定しており，複数のソースドメインがあった場合にソースドメインごとに転移することができず，すべてのソースドメインを同等のものとして扱ってしまうという問題がある．これにより，ターゲットドメインの予測に関係のないソースドメインが混入していると，ターゲットドメインの予測精度が低下する負の転移という現象が発生してしまう．この問題は分類問題に対応した TrAdaBoost でも同様に発生するが，TrAdaBoost の後発の研究で複数ソースからの転移を可能にした手法である MultiSourceTrAdaBoost の提案によって解消されている [10]．MultiSourceTrAdaBoost は弱学習器を生成する際にすべてのソースドメインのデータを一度に使うのではなく，一つずつソースドメインを選択し，選択したソースドメインとターゲットドメインのデータを用いて弱学習器を生成する．ソースドメインごとに生成された弱学習器の中で，ターゲットドメインの予測誤差が最小となったものをそのステップでの弱学習器として採用する．このとき，弱学習器の学習に使用したターゲットドメインのデータを用いて予測誤差を計算するのは不適切であるため，実際には，クロスバリデーションを用いてソースドメインごとに弱学習器を生成して予測誤差を比較する．その後，採用されたソースドメインのデータとすべてのターゲットドメインのデータを用いて生成した弱学習器を，そのステップでの弱学習器として採用している．また，各学習ステップで採用されるソースドメインは一つだけとなる．

アルゴリズム 1 Two-stage TrAdaBoost.R2

Input: データセット T_{source}（データ数 n），T_{target}（データ数 m）を結合した T（入力 $\boldsymbol{x}$，ラベル $\boldsymbol{y}$），ステップ数 S，ブースティング回数 N，クロスバリデーションの数 F，弱学習器 $Learner$，重みの初期値 $\boldsymbol{w}^1(w_i^1 = 1/(n+m)\ 1 \leq i \leq n+m)$)

For $t = 1, \ldots, S$:

1. T, $\boldsymbol{w}^t$, N, $Learner$ を入力とした $AdaBoost.R2'$ モデルである $model_t$ を学習．$AdaBoost.R2'$ モデルは最初の n サンプルの重みを固定したまま学習を進める $AdaBoost.R2$ モデル（$AdaBoost.R2$ のアルゴリズムについては付録を参照されたい）．F-fold クロスバリデーションで T_{target} に対する予測誤差 $error_t$ を取得する．
2. t 時点の $Learner$ と重み $\boldsymbol{w}^t$ を取得し予測値 h_t を得る．
3. 重み調整済み誤差 e_i^t をすべての入力に対して計算．
$$e_i^t = |y_i - h_t(x_i)| / \max_j |y_j - h_t(x_j)|$$
4. 重みを更新
$$w_i^{t+1} = \begin{cases} w_i^t \beta_t^{e_i^t} / Z_t, & 1 \leq i \leq n \\ w_i^t / Z_t, & n+1 \leq i \leq n+m \end{cases}$$
Z_t は正規化のための定数，β_t は T_{target} の重みの総和が $\frac{m}{(n+m)} + \frac{t}{(S-1)}\left(1 - \frac{m}{(n+m)}\right)$ に十分近づくよう二分探索．

Output: $model_t$ where $t = \mathrm{argmin}_t\ error_t$

3. 提案手法

本研究では，ドメイン適応における，複数ソースドメインからの転移を想定した教師あり転移学習の回帰問題を解く手法である，Multi Source Two-stage TrAdaBoost.R2 を提案する．

提案手法は，Two-stage TrAdaBoost.R2 を複数ドメインからの転移を可能にできるように改良した手法である．提案手法のモデル推定法をアルゴリズム 2 に示す．提案手法が，Two-stage TrAdaBoost.R2 と異なる点は次のとおりである．

- 弱学習器生成時，すべてのソースドメインを一度に使用せず，ドメインごとに分けて使用する

アルゴリズム **2**

Multi Source Two-stage TrAdaBoost.R2

Input: ソースドメインの数 K（合計データ数 n）に対応するラベル付きのデータセット $T^k_{source}(k=1,\ldots,K)$, T_{target}（データ数 m），重みの初期値 $\boldsymbol{w}^1_{source}$, $\boldsymbol{w}^{k1}_{target}$（すべて $1/(n+m)$），ステップ数 S，ブースティング回数 N，クロスバリデーションの数 F，弱学習器 $Learner$，

For $t=1,\ldots,S$:

1. T^k_{source}, T_{target}, $\boldsymbol{w}^{kt}_{source}$, $\boldsymbol{w}^t_{target}$, N, L を入力として $model_t$ を学習．このとき，各ブースティングステップ j で取得する $Learner^j_t$, $error^j_t$ $(j=1,\ldots,N)$ は以下のように導出する．
   （ア）T_{target}, $\boldsymbol{w}^t_{target}$ を入力として F-fold クロスバリデーションを用いて，T_{target} に対する予測誤差 $error_0$ を得る．
   （イ）*For* $k=1,\ldots,K$:
   T^k_{source}, $\boldsymbol{w}^{kt}_{source}$, T_{target}, $\boldsymbol{w}^t_{target}$ を入力として F-fold クロスバリデーションを用いて，T_{target} に対する予測誤差 $error_k$ を得る．
   （ウ）$error_k < error_0$ となった T^k_{source}, T_{target} とそれに対応する重みを用いて弱学習器 $Learner^j_t$ を学習．同様に F-fold クロスバリデーションを用いて T_{target} に対する予測誤差 $error^j_t$ を得る．
2. $error_t$ を求める　$error_t = \frac{1}{N}\sum_j error^j_t$
3. t 時点での $Learner$ と重み $\boldsymbol{w}^t$ を取得し予測値 h_t を得る．
4. 重み調整済み誤差 e^t_i をすべての入力に対して計算．
5. 重みを更新

$$w^{t+1}_i = \begin{cases} w^t_i \beta_t^{e^t_i}/Z_t, & 1 \le i \le n \\ w^t_i/Z_t, & n+1 \le i \le n+m \end{cases}$$

Z_t は正規化のための定数，β_t は T_{target} の重みの総和が $\frac{m}{(n+m)} + \frac{t}{(S-1)}\left(1-\frac{m}{(n+m)}\right)$ に十分近づくよう二分探索．

Output: $model_t$ where $t = \mathrm{argmin}_t\ error_t$

- $error_0$ より誤差の小さい $error_k$ の生成に寄与したすべてのソースドメインとターゲットドメインを用いて再度弱学習器 $Learner^j_t$ を生成する
- クロスバリデーションを用いて $Learner^j_t$ の予測誤差 $error^j_t$ を算出する．$error^j_t$ から強学習器 $model_t$ の予測誤差 $error_t$ を算出する

弱学習器生成時に，ソースドメインごとに分割するアプローチは MultiSourceTrAdaBoost と非常に似ている．しかし，提案手法ではターゲットドメインのみを用いて生成した弱学習器と各ソースドメインも使用して生成した弱学習器の予測誤差を比較しているため，採用するソースドメインの数を動的に決定することができる．予測誤差が最小になったただ一つのソースドメインのみを採用する MultiSourceTrAdaBoost では，このような柔軟なドメインの選択を行えないため，転移可能なソースドメインを取りこぼす可能性がある．

表 1　使用データ例

StatusTime	Radio Number	Latitude	Longitude	Vehicle Status
2016-04-01 00:28:01	145392	35.713585	139.792383	空車
2016-04-01 00:28:01	145392	35.692049	139.769380	実車

表 2　明治神宮・浅草寺周辺のタクシー需要量

	1 月 1 日	1 月 8 日
明治神宮周辺	963	838
浅草寺周辺	545	354

また，クロスバリデーションを適用するタイミングについても提案手法は工夫を行っている．Two-stage TrAdaBoost.R2 は複数個生成した学習器 AdaBoost.R2 から最良のものを選択するためにクロスバリデーションを用いている．この方法を提案手法にも適用すると，各学習ステップで弱学習器の予測誤差を算出するクロスバリデーションをネストする形になってしまうため，計算コストが増大する．そこで提案手法では，弱学習器の予測誤差を用いて学習器 AdaBoost.R2 の予測誤差を算出することで，学習器の選択の際にクロスバリデーションを使用せず，計算コストを抑えることに成功した．

4. データ概要

本研究では，経営科学系研究部会連合協議会主催，令和元年度データ解析コンペティションにて提供された東京と周辺を走行するタクシーのプローブデータを使用した．記録されているレコードの例を表 1 に示す．

本研究では，タクシーの状態が空車などの状況から実車に変化したレコードを乗車ポイントとし，需要点と定義する．具体的には，車輌識別番号である Radio-Number ごとに VehicleStatus の変化に基づいて需要点を抽出する．

5. 実験方法

本研究では，元旦の明治神宮・浅草寺周辺のタクシー需要量を予測する．需要量は需要点の個数を意味する．表 2 は，2018 年の 1 月 1 日とその翌週の 1 月 8 日における明治神宮周辺 3 km × 3 km と浅草寺周辺 2 km × 2 km の深夜 0 時から 5 時にかけてのタクシー需要量である．このデータを見てもわかるように，元日深夜のタクシー需要はその翌週の通常期である 1 月 8 日と比べて大きくなっている．明治神宮および浅草寺に

は初詣の来場者が多く，電車などの公共交通機関が深夜でも動いており，通常期とは異なる需要が発生していることが需要増加の原因であると考えられる．なお，本研究では予測対象を深夜の時間帯に限定せず，1 日全体としている．これは，予測の評価を行う検証データのサンプル数を確保し，モデルの精度を適切に検証するためである．

提供データのうち明治神宮周辺 3 km × 3 km と浅草寺周辺 2 km × 2 km のエリアを抽出し，予測モデルの学習に 2016 年 12 月，2017 年 1 月 1 日，2017 年 12 月を使用し，検証対象に 2018 年 1 月 1 日のデータを使用した．需要点の件数は，学習データが約 36 万件，検証データが約 6,000 件となっている．

また，小範囲ごとのタクシー需要量を予測するために，抽出した需要点のレコードを密度クラスタリングの手法である Density Peaks Clustering (DPC) [11] を用いてクラスタリングを行う．密度に基づくクラスタリングを行う DPC を使用することによって，非円状のクラスター構造を検知し，道路の構造に即したクラスタリングが可能となる．そのため，k-means などの他の距離に基づくクラスタリング手法と比べ，より自然な結果を得ることができる．なお，日ごとのデータそれぞれに DPC を適用すると，毎回クラスターの形状が変わってしまうため，日をまたいでクラスタリング結果を用いることができないという課題があった．そこで本研究では，基準となる日の需要点データを用いて DPC でクラスタリングを行い，それ以外の日の各需要点について，基準日の需要点との距離が最小になるように基準日の需要点が属するクラスターに割り当てるという方法によるクラスタリング結果を用いた．今回の実験では基準日を 2017 年 1 月 1 日とし，明治神宮周辺が 12 個のクラスターに，浅草寺周辺が 11 個のクラスターにそれぞれ分割された．明治神宮周辺のクラスタリング結果を図 1，浅草寺周辺のクラスタリング結果を図 2 に示す．図 1，図 2 のクラスタリング結果を見ると，どのクラスターもおおよそ道に沿った形状になっていることがわかる．

クラスターごとに 1 時間当たりのタクシー需要量を集計し，需要予測モデルの目的変数とする．特徴量には時刻，日付，クラスター番号を使用し，クラスター番号と日付の組合せをドメインとする．たとえば 2018 年 1 月 1 日クラスター番号 1 の地域の需要量を予測する場合，2017 年 1 月 1 日クラスター番号 1 番の需要量に関するデータをターゲットドメインとし，12 月 24 日クラスター番号 1 のドメインや 2017 年 1 月 1 日クラスター番号 2 のドメインなど，それ以外すべての学習データをソースドメインとして学習を行う．なお，2016 年 12 月と 2017 年 12 月のデータは同じ日付かつ同じクラスター番号のデータを同一ドメインとして扱う．

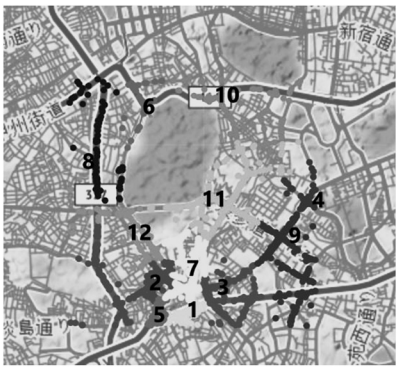

図 1　需要点のクラスタリング結果（明治神宮周辺）

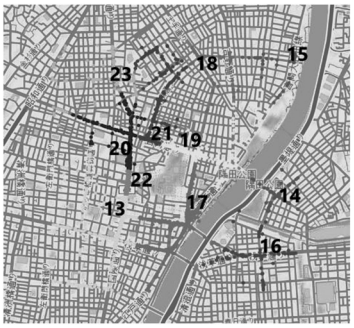

図 2　需要点のクラスタリング結果（浅草寺周辺）

表 3　各手法とデータセットの対応表

手法	学習データ		検証データ
	Target	Source	
AdaBoost all	○	○	Target
AdaBoost target	○	×	Target
Two-stage TrAdaBoost.R2	○	○（同一）	Target
Multi Source Two-stage TrAdaBoost.R2	○	○（区別）	Target

予測精度の比較対象として，Two-stage TrAdaBoost.R2 以外に，ソースドメインとターゲットドメインを区別せず，すべてのデータを用いて学習した AdaBoost.R2 (AdaBoost all)，ターゲットドメインのみを用いて学習した AdaBoost.R2 (AdaBoost target) を採用した．各手法で用いるデータセットの対応表を表 3 に示す．ここではターゲットドメインを Target，ソースドメインを Source と表現する．また，Two-

stage TrAdaBoost.R2 ではすべてのソースドメインをそれぞれ区別せずに同一のものとして扱うが，Multi Source Two-stage TrAdaBoost.R2 は各ソースドメインを区別して扱う．

すべてのモデルにおいて，弱学習器には決定木 (CART [12]) を採用している．各ハイパーパラメータはブースティング回数 $N = 50$，ステップ数 $S = 10$，クロスバリデーションの Fold 数 $F = 5$ となっている．また，決定木の最大深度は 6 とした．

2018 年 1 月 1 日のクラスターごとの 1 時間当たりのタクシー需要量を予測し，予測精度の評価指標には RMSE を採用した．

6. 実験結果・考察

5 節で示した方法で定義したクラスターごとに 1 時間当たりのタクシー需要量を予測したところ，各手法の RMSE 値は表 4 のとおりになった．提案手法である Multi Source Two-stage TrAdaBoost.R2 が他の比較手法を抑え，最良の予測精度を記録した．

この結果より，複数のソースドメインデータからの転移を想定した回帰問題において，提案手法が最も当てはまりがよいことが示された．

また，本研究では明治神宮・浅草寺周辺で通常とは異なる需要が発生していると仮説を立てているため，それらの施設付近のクラスターでの予測結果に特に注目するべきである．明治神宮の入り口（原宿口）と浅草寺の入り口（雷門）に隣接するクラスターでの予測結果は表 5 のようになった．表 5 の結果を見ると，全体の予測結果よりもさらに差をつけて提案手法がよい精度を記録していることがわかる．このことから，通常とは異なる需要が発生している明治神宮・浅草寺周辺での予測において特に，提案手法の有効性が示された．

同様に，元旦の深夜の時間帯では通常動いていない公共交通機関が動いているのもあり，通常時とは異なる需要が発生していると考えられる．そこで，0 時から 5 時までの予測結果を表 6 に示す．深夜の時間帯の予測精度を比較すると，全体の予測結果よりもさらに差をつけて提案手法が最も高い予測精度を記録した．通常，公共交通機関が動いていない時間帯においての予測でも，提案手法の有効性が示された．

ここで挙げた二つの条件は，特に通常時と異なる性質をもった需要が発生しているため，転移学習が必要な状況であると考えられる．これらの条件下において提案手法が最も高い予測精度を記録したことから，提案手法は適切なソースドメインからの転移を行うことに成功したと考えられる．

表 4　手法ごとの予測精度

手法	RMSE
AdaBoost all	11.099
AdaBoost target	6.673
Two-stage TrAdaBoost.R2	6.693
Multi Source Two-stage TrAdaBoost.R2	**5.939**

表 5　施設付近クラスターの予測精度 (RMSE)

手法	明治神宮	浅草寺
AdaBoost all	16.45	10.27
AdaBoost target	11.77	11.50
Two-stage TrAdaBoost.R2	11.67	11.49
Multi Source Two-stage TrAdaBoost.R2	**7.76**	**8.79**

表 6　0 時から 5 時の予測精度 (RMSE)

手法	明治神宮	浅草寺
AdaBoost all	12.42	6.97
AdaBoost target	14.83	7.65
Two-stage TrAdaBoost.R2	13.86	7.15
Multi Source Two-stage TrAdaBoost.R2	**11.89**	**5.01**

7. おわりに

本研究では，複数ソースドメインからの転移を想定した教師付き転移学習アルゴリズム Multi Source Two-stage TrAdaBoost.R2 を提案した．タクシーのプローブデータを使用し，元日における，明治神宮・浅草寺周辺の 1 時間当たりのタクシー需要量を予測したところ，AdaBoost.R2 や Two-stage TrAdaBoost.R2 を上回る予測精度を記録した．特に，上記施設付近および深夜の時間帯に関しては，提案手法が他の手法に差をつけて高い予測精度を記録した．これらの条件では，より一層通常期と異なる性質をもった需要が発生していると考えられるため，適切なソースドメインからの転移を可能にした提案手法の強みが活かされた結果であるといえる．

しかし，提案手法 Multi Source Two-stage TrAdaBoost.R2 は精度面で良好な結果を残したものの，既存手法 Two-stage TrAdaBoost.R2 と比べて計算量が多くなるという課題がある．各ブースティングステップで有効なドメインを選択するため，おおよそソース

ドメインのドメイン数倍だけ計算量が既存手法から増加する．ドメインを細かく分割すると精度向上を望めるが，計算量が増加するため，運用時間を考慮して適切な分割方法を設定する必要がある．

また，本研究をより実用面に特化させる方針として，次の二つが挙げられる．これらは今後の課題としたい．

- 過去の需要量などの特徴量を追加し，さらなる予測精度の向上を目指す
- 潜在的な需要を考慮した予測モデルに発展させる

謝辞　2 名の査読者の方には，論文を丁寧に読んでいただき，多くの貴重なコメントをいただきました．厚く御礼申し上げます．

参考文献

[1] P. Friederichs and T. L. Thorarinsdottir, “Forecast verification for extreme value distributions with an application to probabilistic peak wind prediction,” *Environmetrics*, **23**, pp. 579–594, 2012.

[2] アクセンチュア,「トヨタ，JapanTaxi，KDDI，アクセンチュアの 4 社，人工知能を活用したタクシーの「配車支援システム」の試験導入を開始」，https://www.accenture.com/jp-ja/company-news-releases-20180309?src=PSEARCH（2020 年 6 月 29 日閲覧）

[3] デンソーテン,「人工知能を活用したタクシー需要リアルタイム予測 AI 需要予測サービス 」，https://www.denso-ten.com/jp/c-system/ai/index.html（2020 年 6 月 29 日閲覧）

[4] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, Z. Li, J. Ye and D. Chuxing, “Deep multi-view spatial-temporal network for taxi demand prediction,” In *Proceeding of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 2588–2595, 2018.

[5] H. Drucker, “Improving regressors using boosting techniques,” In *Proceeding of the 14th International Conference on Machine Learning*, pp. 863–870, 1997.

[6] D. Pardoe and P. Stone, “Boosting for regression transfer,” In *Proceedings of the 27th International Conference on Machine Learning*, pp. 863–870, 2010.

[7] J. Horne and W. Manzenreiter, “Accounting for mega-events: Forecast and actual impacts of the 2002 football world cup finals on the host countries Japan/Korea,” *International Review for Sociology of Sport*, **39**, pp. 187–203, 2004.

[8] I. Redko, E. Morvant, A. Habrard, M. Sebban and Y. Bennani, *Advances in Domain Adaptation Theory*, Elsevier, 2019.

[9] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, “Boosting for transfer learning,” In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[10] Y. Yao and G. Doretto, “Boosting for transfer learning with multiple sources,” In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[11] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, **34**, pp. 1492–1496, 2014.

[12] A. D. Gordon, L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, “Classification and regression trees,” *Biometrics*, **40**, p. 874, 1984.

付 録

アルゴリズム (ア)　AdaBoost.R2

Input: データセット T（データ数 n），ブースティング回数 N, 弱学習器 *Learner*, 重みの初期値 $\boldsymbol{w}^1$ ($w_i^1 = 1/n \quad 1 \leq i \leq n$)

For $t = 1, \ldots, N$:

1. T, $\boldsymbol{w}^t$ を用いて学習した *Learner* を h_t とする
2. 各データに対して調整誤差 e_i^t を計算する

$$D_t = \max_j |y_i - h_t(x_j)| \quad (j = 1, \ldots, n)$$

$$e_i^t = \frac{|y_i - h_t(x_i)|}{D_t}$$

3. h_t の調整誤差 ϵ_t を計算する
   $\epsilon_t \geq 0.5$ の場合は For 文処理を抜けて $N = t - 1$ とする

$$\epsilon_t = \sum_{i=1}^{n} e_i^t w_i^t$$

4. $\beta_t = \epsilon_t/(1 - \epsilon_t)$ を計算する
5. 重みを下記の式で更新する

$$w_i^{t+1} = w_i^t \beta_t^{1-e_i^t}/Z_t$$

Z_t は正規化のための定数

Output: $h_f(\boldsymbol{x}) = \ln(1/\beta_t)$ で重みづけした $h_t(\boldsymbol{x})$ ($t = 1, 2, \ldots, N$) の加重中央値 ($1 \leq t \leq N$)