

電力中央研究所 研究資料

NO. Y19511

アンサンブル学習の予測有効性について
—電力需要予測を対象とした検証—

2020年3月

一般財団法人 電力中央研究所

IR

CRIEPI

Central Research Institute of
Electric Power Industry

アンサンブル学習の予測有効性について —電力需要予測を対象とした検証—

加部 哲史^{*1}

^{*1}社会経済研究所 事業制度・経済分析領域 主任研究員

背景

近年、訓練用データをもとにモデルを構築し、未知データを予測するために機械学習を用いたデータ解析に注目が集まっている。その中で、複数の学習器（弱学習器）¹⁾について訓練を行い、それらを結合させることで、予測精度の改善を図る「アンサンブル学習」という方法が活用されている。

目的

アンサンブル学習を電力需要予測に適用し、単一学習器との予測精度を比較することで、アンサンブル学習の特徴を把握するとともに、その有効性について検証を行う。なお、本研究では、予測精度そのものを対象とせず、予測精度の比較に焦点を当てる。

主な成果

アンサンブル学習（とそのアルゴリズム）の概要

アンサンブル学習は、複数の学習器（弱学習器）について訓練を行い、それらを結合させ、予測精度に優れた学習器（強学習器）に近づけるための学習アルゴリズムである（図 1）。有名なアルゴリズムとして、バギング（Bagging）とブースティング（Boosting）の 2 つが挙げられる。バギングでは、統計学で用いられるブートストラップ法を用いて、訓練用データの部分集合をブートストラップ標本として生成することで、複数の弱学習器を生成し、それらを結合することで予測を行う（図 2）。一方、ブースティングは、従前の学習結果をもとに、弱学習器の訓練方法を変えることで、全体の予測結果を改善していく逐次アルゴリズムである（図 3）。

電力需要予測の精度比較

本研究では、需要予測に電力広域的運営推進機関で公表されている 1 時間毎の電力需要（MW、東京エリア、2016/4/1～2019/3/31）と気象庁が公表している 1 時間毎の気象データ（東京）を用いる。また、先行研究と同様に、弱学習器としてニューラルネットワークを採用し、アンサンブル学習を試みる。アンサンブル学習による予測結果を表 1 に示す。

表 1 より、アンサンブル学習を適用することで、単一学習器（Single）と比較して、予測誤差（RMSE, Root Mean Squared Error）の改善がみられた。また、月別、曜日別、時間別に予測誤差の分布をみると、アンサンブル学習を用いた予測の方が単一学習器よりも、誤差のばらつきが小さくなり、予測精度の改善がみられた。特に、単一学習器の予測誤差が大きくなる時点で、改善傾向が顕著にみられた。

¹⁾ 学習器とは、訓練用データから学習されたモデルを指す。

本研究で得られた示唆

今回の分析から、アンサンブル学習を適用することで、単一学習器では予測が難しい時点でも、複数の学習器を組み合わせることで、予測誤差が改善する可能性が考えられる。ただし、多くの弱学習器が必要となる場合には、単一学習器よりも計算負荷が大きくなる点に留意する必要がある。さらに、アンサンブル学習では複数のモデルを集約するため、モデルの解釈性を欠くことになり、これらの問題を改善していくことが、今後重要となる。

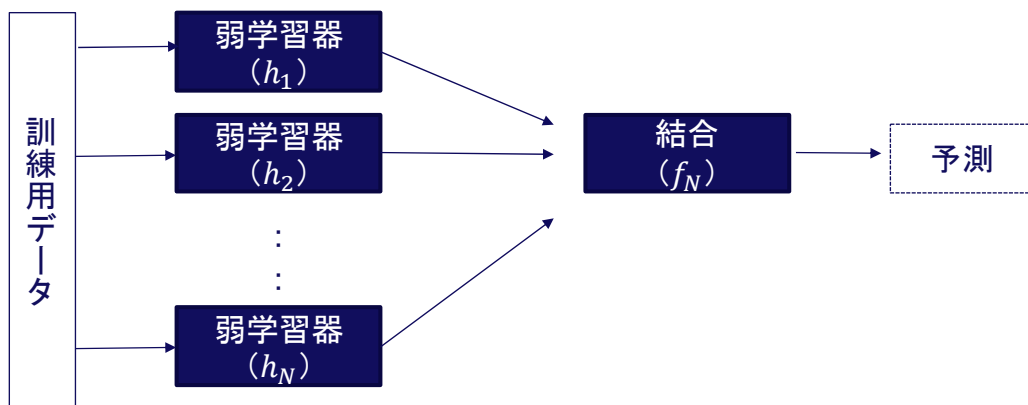


図1 アンサンブル学習の概要

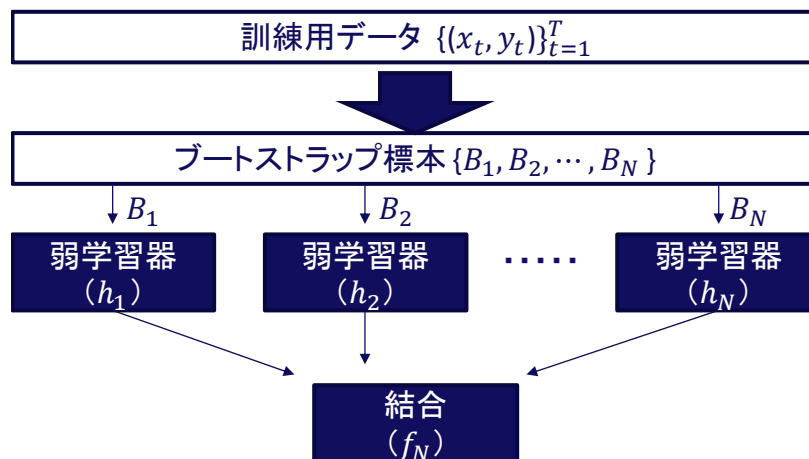


図2 バギングアルゴリズムの概要

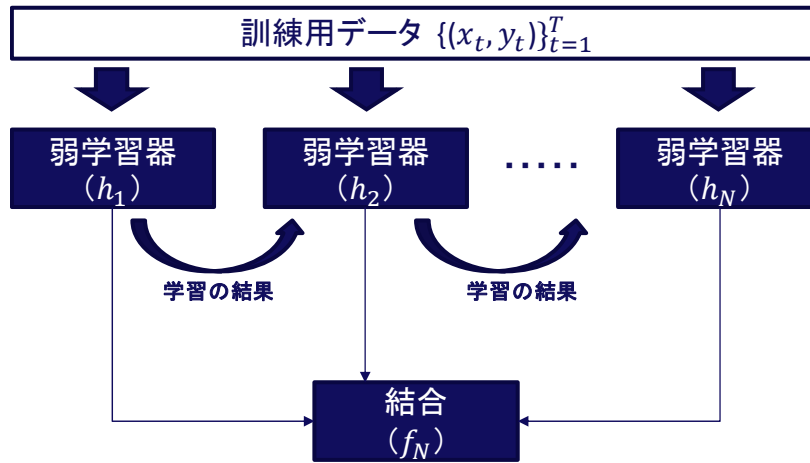


図3 ブースティングアルゴリズムの概要

表1 予測誤差の比較（東京エリア、期間 2018/4/1～2019/3/31）

学習率 パラメータ		バギング			ブースティング		
		Single	Bagging	Bagging.SE	AdaBoost.R	AdaBoost.R2	GdBoost
0.25	RMSE	1967	1910	1909	1818	1918	1938
	増減率		-2.9%	-2.9%	-7.6%	-2.5%	-1.5%
	学習器の数		7	5	9	8	3
0.50	RMSE	1939	1848	1743	1730	1687	1901
	増減率		-4.7%	-10.1%	-10.8%	-13.0%	-2.0%
	学習器の数		10	2	5	2	7
0.75	RMSE	2070	2004	1991	1931	1993	1995
	増減率		-3.2%	-3.8%	-6.7%	-3.7%	-3.6%
	学習器の数		7	2	3	2	6

- ・東京エリアは、栃木県、群馬県、茨城県、埼玉県、千葉県、東京都、神奈川県、山梨県、静岡県の一部。
- ・増減率は、単一学習器 (Single) に対する RMSE (Root Mean Squared Error) の変化を表す。
- ・ニューラルネットワークの訓練回数(は 1000 回、勾配法の学習率パラメータは 3 パターン {0.25, 0.50, 0.75} を採用。
- ・アンサンブル学習における弱学習器の数は RMSE が最小になるものを採用、ただし最大 10 個まで $\{h_1, h_2, \dots, h_{10}\}$ 。
- ・Bagging. SE は、Bagging に選択的アンサンブルを適用したケースを指す。

関連報告書：

[1]Y17002 「JEPX スポット市場における価格の長期予測手法の検討」(2018.3)

目 次

1. はじめに	1
2. アンサンブル学習とは	1
2.1 アンサンブル学習の利点	3
2.2 バギング (Bagging) アルゴリズムの概要	4
2.3 ブースティング (Boosting) アルゴリズムの概要	4
2.4 選択的アンサンブルとは	6
3. ニューラルネットワークを用いたアンサンブル学習について	7
3.1 先行研究	7
3.2 ニューラルネットワークとは	7
3.3 誤差逆伝搬法を用いたパラメータ推定	8
4. 電力需要予測の精度比較	9
4.1 データの概要	9
4.2 ニューラルネットワークの設定	9
4.3 予測精度の比較結果	10
5. 考察とまとめ	11
参考文献	11
付録 A 予測誤差の分布	13

1. はじめに

近年、IT技術の発展に伴い、コンピュータを介して大量に蓄積された大規模データの活用が様々な企業や組織で行われている。回帰分析のようなこれまでの統計学や計量経済学におけるデータ分析は非常に有用であるものの、大規模データの中から意味のあるシグナルとノイズを選別する手法¹⁾として、機械学習が注目されている。

機械学習とは、日々人間が行っているパターン認識や経験則を導き出す活動を、コンピュータを使って実現するための技術やアルゴリズム、あるいはソフトウェアなどの総称である。具体的には、機械学習では一定数以上の訓練用データから、目的に応じてアルゴリズムを選択し、データの中から規則性や法則性を自動的にコンピュータに学習させる。

機械学習は、主に教師あり学習と教師なし学習の2種類に分類される。教師あり学習では、事前に与えられた訓練用データをいわば「例題（＝教師からの助言）」とみなして、それをもとに学習を行う。例えば、統計学や計量経済学で頻繁に使われる回帰分析は、教師あり学習に分類される。

一方、教師なし学習では、観測されたデータから特徴量を学習し、それに従って、パターンやカテゴリに自動分類を行う。例えば主成分分析や因子分析がこれにあたる。

機械学習が取り扱う問題として主に、1) 分類、2) 回帰、3) クラスタリング、4) ルールの抽出などが挙げられる。1) 分類では、観測データ y にラベルが付けられ、機械学習を通して、ラベルのない新しいデータに、ラベルを割り当てるかどうかの判断を下す。2) 回帰では、ラベルの代わりに観測データ y が持つ数値を用いて、観測データを説明するための関数 $y = f(x)$ を推定する。新たな変数 x_0 が与えられたとき、推定された関数

\hat{f} を用いて、予測値 $\hat{y} = \hat{f}(x_0)$ が得られることになる。3) クラスタリングでは、ラベルの付与無しに、観測データの類似度を学習し、クラスター（塊）にグループ分けを行う。4) ルールの抽出では、データ間の相関関係に基づき、自動的に観測されたデータからルールの抽出を行う。ここで1) 分類と2) 回帰は教師あり学習に分類され、3) クラスタリングと4) ルールの抽出は教師なし学習に分類される。

機械学習により、コンピュータは観測されたデータから反復的に学習し、自律的に大量のデータに潜むパターンを見つけ出すことができる。さらに、学習した結果を新たなデータに当てはめることで、パターンに従って、予測を行うことができるため、様々な分野で応用されている。

機械学習を用いたデータ解析に注目が集まる中で、本研究ではアンサンブル学習という学習方法を採用し、電力需要の予測精度を比較することで、アンサンブル学習の有効性について検証を行う。本研究は、予測精度そのものを対象とせず、予測精度の比較に焦点を当てたものである。

以下では、第2章でアンサンブル学習の概要について述べる。次に、第3章では電力需要を予測するために用いるニューラルネットワークについて説明する。第4章では、ニューラルネットワークを用いたアンサンブル学習を行い、学習方法の違いによる予測精度の比較を行う。最後に、第5章で分析結果について考察を行う。

2. アンサンブル学習とは

アンサンブル学習は、同じ訓練用データから1つの学習器を構築する通常のアプローチとは異なり、複数の学習器（弱学習器）²⁾を生成し、それらを結合することで予測精度の向上を試みる学習アルゴリズムである（図1）。有名なアルゴリズム

¹⁾ 機械学習には、伝統的な統計分析も多く含まれる。

²⁾ 学習器とは、訓練用データから学習されたモデル。

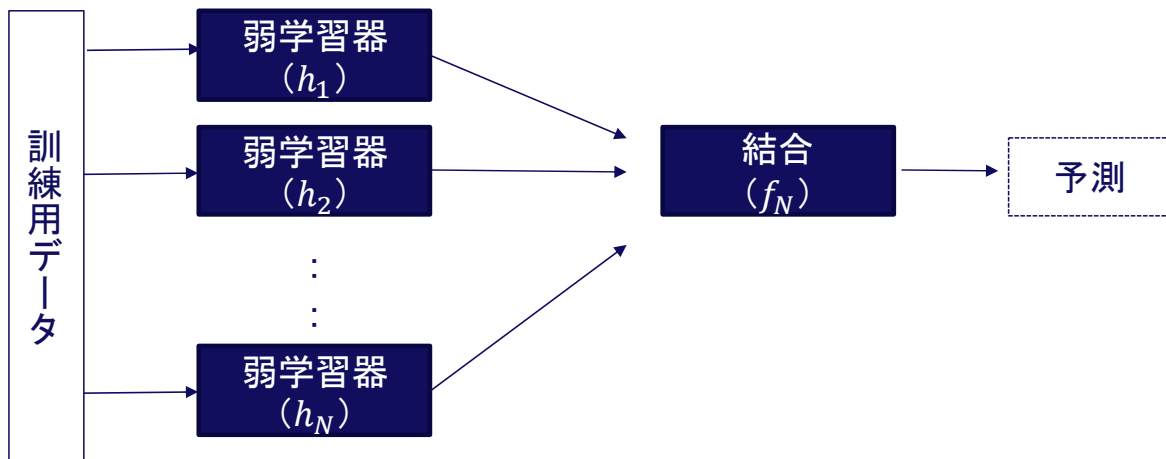


図1 アンサンブル学習の概要

ムとして、バギング (Bagging) とブースティング (Boosting) の 2 つが挙げられる。これらのアルゴリズムに関しては、2.2 節と 2.3 節で説明する。

アンサンブル学習の簡単な例として、平均化 (averaging) について考える。平均化は、数値データに対する最もよく用いられる基本的な結合法である。ここでは、予測に関してどのように平均化が機能するのかを説明するために、回帰問題を例にとる。

N 個の独立に生成された弱学習器の集合 $\{h_1, h_2, \dots, h_N\}$ が与えられ、説明変数 x に対する $h_i(x)$ の出力は $h_i \in \mathbb{R}$ であるとするとき、単純平均によって結合された関数 $f_N(x)$ は以下のように表される。

$$f_N(x) = \frac{1}{N} \sum_{i=1}^N h_i(x) \quad (1)$$

さらに、真の関数を $f^*(x)$ 、 x を分布 $p(x)$ に従い抽出された標本と仮定するとき、弱学習器 $h_i(x)$ の二乗誤差の期待値は以下で表される。

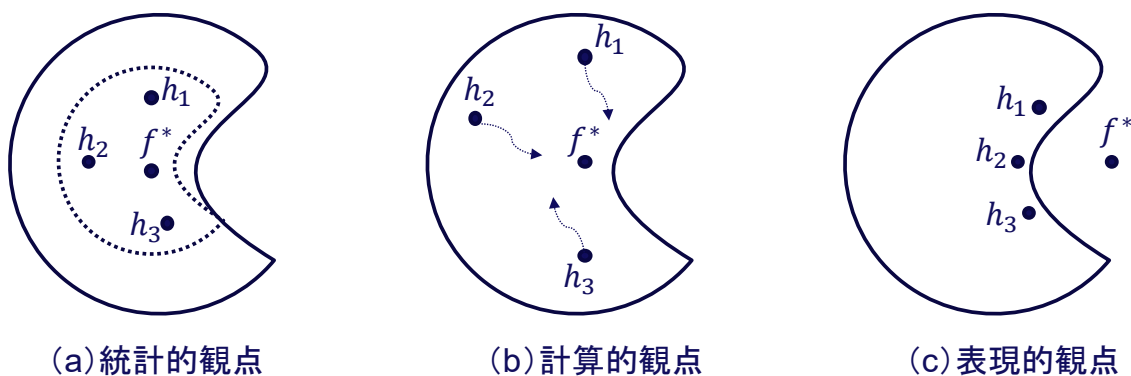
$$\begin{aligned} err(h_i) &\equiv \int (h_i(x) - f^*(x))^2 p(x) dx \\ &= \int \varepsilon_i(x)^2 p(x) dx \end{aligned} \quad (2)$$

このとき、全ての弱学習器 $\{h_1, h_2, \dots, h_N\}$ に対して平均値をとると以下ようになる。

$$\frac{1}{N} \sum_{i=1}^N err(h_i) = \frac{1}{N} \sum_{i=1}^N \int \varepsilon_i(x)^2 p(x) dx \quad (3)$$

また、単純平均により結合された関数 $f_N(x)$ についての二乗誤差の期待値は以下のように示せる。

$$\begin{aligned} err(f_N) &\equiv \int \left(\frac{1}{N} \sum_{i=1}^N h_i(x) - f^*(x) \right)^2 p(x) dx \\ &= \int \left(\frac{1}{N} \sum_{i=1}^N \varepsilon_i(x) \right)^2 p(x) dx \end{aligned} \quad (4)$$



出典：Dietterich (2000)、Zhou(2012)

注) 図中の実線は関数空間を表す。(a)の破線はモデル集合を表す。(b)の矢印は初期値からの局所探索を表す。

図2 アンサンブル学習の利点

したがって、式(3)と式(4)より次の不等式が成り立つことが分かる³⁾。

$$err(f_N) \leq \frac{1}{N} \sum_{i=1}^N err(h_i) \quad (5)$$

このことから、理論的には単純平均により結合された関数 $f_N(x)$ の二乗誤差の期待値 $err(f_N)$ は、個々に評価された場合 (i.e., $err(h_i)$) の平均値を下回るか少なくとも同程度の精度で予測できることが分かる。

2.1 アンサンブル学習の利点

Dietterich (2000) と Zhou (2012) は、アンサンブル学習の利点として、以下の3つの観点から考察を行っている (図2参照)。

(a) 統計的観点

同じ訓練用データに対して、同様な正確さを与える幾つかの異なったモデルが存在する場合がある。これらの中から1つを選ん

だ場合、それが未知のデータに対して良い予測ができない間違っただモデルであるリスクが存在する。そこで、複数のモデルを結合することによって、間違っただモデルを選択してしまうリスクを減らすことができる。

(b) 計算的観点

多くのモデルは、ある種の局所探索を行うので、局所的な最適解にとどまってしまうことがある。訓練用データが十分にある場合でも、最適なモデルを見つけることは非常に難しい。そこで、多くの異なる初期値から局所探索を行い、複数のモデルを結合することによって、間違っただ極小値 (or 極大値) を選択してしまうリスクを減らすことができる。

(c) 表現的観点

多くの機械学習の実行において、真のモデルは関数空間内の任意の関数では表現できないことがある。そこで、複数のモデルを結合することによって表現可能な関数の空

³⁾ 標本分散の定義より、以下の不等式を利用する。

$$\text{var}(x) = \frac{1}{N} \sum x_i^2 - \bar{x}^2 \geq 0$$

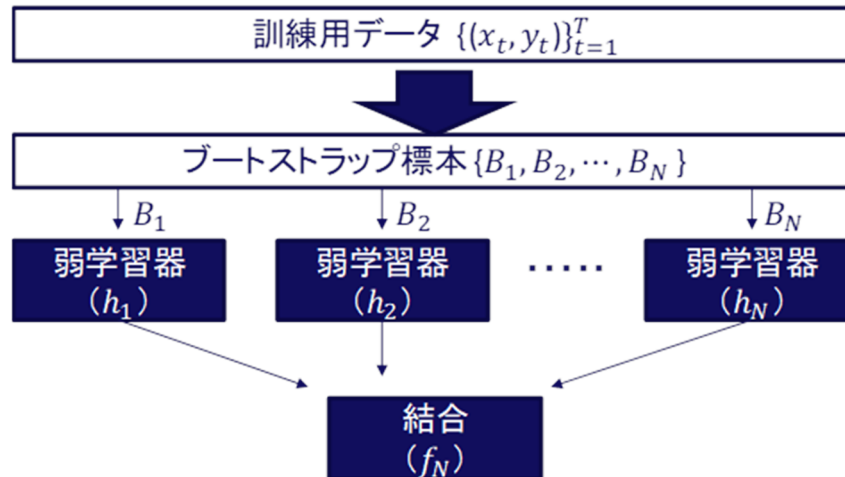


図3 バギングアルゴリズムの概要

間を広げることが可能であり、それにより真のモデルに対するより良い近似を形成することができる。

2.2 バギング (Bagging) アルゴリズムの概要

回帰問題に関する平均化の例では、独立に生成された弱学習器から得られた予測結果を結合することで、二乗誤差の期待値が軽減される可能性を示した。ただし、複数の弱学習器 $\{h_1, h_2, \dots, h_N\}$ を生成するためには、複数の訓練用データを用意する必要がある。バギング (Breiman 1996)⁴⁾ では、統計学で用いられるブートストラップ法を用いて、訓練用データの部分集合をブートストラップ標本 $\{B_1, B_2, \dots, B_N\}$ から生成することで、複数の弱学習器を構築する (図3)。

例えば、標本数 T の訓練用データ $\{(x_t, y_t)\}_{t=1}^T$ が与えられた場合、元データから m 個の標本を復元抽出する。この作業を N 回繰り返すことで、 N 個

のブートストラップ標本 $\{B_1, B_2, \dots, B_N\}$ が生成される。一般的に、回帰問題に関しては、平均化による弱学習器の結合を行う。

2.3 ブースティング (Boosting) アルゴリズムの概要

ブースティングは、バギングと並んでよく使われるアルゴリズムである。バギングでは、弱学習器 $\{h_1, h_2, \dots, h_N\}$ は独立しており、相互に関係していない並列アルゴリズムである。一方、ブースティングは、従前の学習結果をもとに、弱学習器の訓練方法を変えることで、全体の予測結果を改善していく逐次アルゴリズムである (図4参照)。ブースティングには、主に AdaBoost (Freund & Schapire 1997) と勾配ブースティング (Friedman 2001) という2種類のア​​ルゴリズムがある。

Freund & Schapire (1997) では、はじめに二値分類問題について AdaBoost アルゴリズムを構築し、その後、多値分類問題については、AdaBoost.M1

⁴⁾ Bagging という名前は Bootstrap Aggregating の略称からきている。

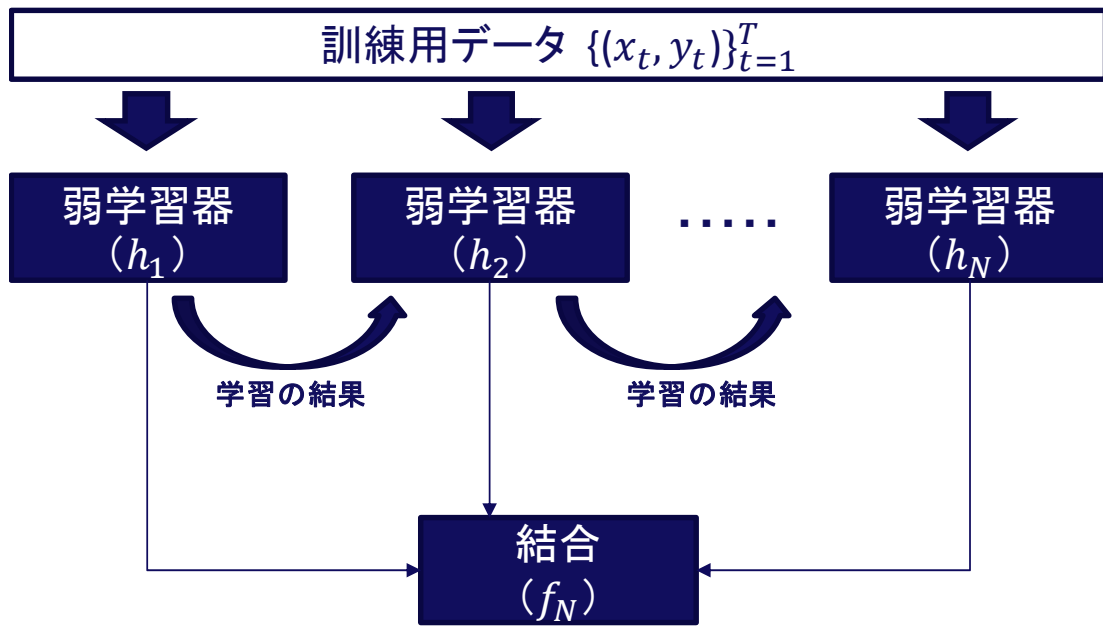


図4 ブースティングアルゴリズムの概要

と AdaBoost.M2、回帰問題については AdaBoost.R を構築した。また、Drucker (1997) は、回帰問題について AdaBoost.R を改良した AdaBoost.R2 を提案している⁵⁾。

AdaBoost では、全てのデータに重みが設定され、訓練誤差の大きい事例については重みを増やし学習することで、予測精度を改善させる。重みは事前の学習結果に基づき更新される。

Freund & Schapire (1997) は、AdaBoost によって構築された弱学習器 $\{h_1, h_2, \dots, h_N\}$ について評価された誤差 $\{\epsilon_1, \epsilon_2, \dots, \epsilon_N\}$ が与えられたとき、最終的に結合される関数 $f_N(x)$ の訓練誤差の上界⁶⁾を示した。例えば、回帰問題に関する訓練誤差の上界は、以下のようになる。

$$\begin{aligned}
 E_{t \sim D} [(f_N(x_t) - y_t)^2] & \\
 & \leq 2^N \prod_{i=1}^N \sqrt{\epsilon_i(1 - \epsilon_i)} \\
 & \leq \exp \left\{ -2 \sum_{i=1}^N \gamma_i^2 \right\} \quad (6)
 \end{aligned}$$

(証明略)

ここで優位度 $\gamma_i = 0.5 - \epsilon_i (\geq 0)$ と仮定する。また、 D は訓練用データの確率分布を表す。このとき、AdaBoost は訓練誤差を指数関数的に減少させることが分かる (i.e., AdaBoost のブースティング性)。

勾配ブースティング (Friedman 2001) では、確率的勾配降下法に基づいて、損失関数 $L(y, f)$ が小さくなるように、弱学習器 $\{h_1, h_2, \dots, h_N\}$ を結合した関数 $f_N(x)$ の更新を行う。

⁵⁾ AdaBoost.R2 では、各データの重みを計算するために必要となる損失評価の方法が簡便になっている。

⁶⁾ AdaBoost では、 $\epsilon_i > 0.5$ となる学習器は取り除かれる。そのため、式(6)が成り立つためには、優位度 $\gamma_i \geq 0$ を

全ての学習器に仮定する必要がある。また、 ϵ_i は誤判別率で、ランダムでは 0.5 となり、ランダムよりどの程度、誤判別率が低いかを優位度で示している。

損失関数 $L(y, f(x))$ が与えられたとき、確率的勾配降下法により関数 f_N は以下のように更新される。

$$f_N(x) \leftarrow f_{N-1}(x) - \rho \left. \frac{\partial L(y_t, f(x_t))}{\partial f(x_t)} \right|_{f=f_{N-1}} \quad (7)$$

ここで、例えば損失関数 $L(y, f) = \frac{1}{2}(y - f)^2$ 、学習率パラメータ $\rho = 1$ と設定するとき、 N 番目の弱学習器 h_N を用いて以下のように表される。

$$\begin{aligned} f_N(x) &\leftarrow f_{N-1}(x) + (y_t - f_{N-1}(x_t)) \\ &= f_{N-1}(x) + h_N(x_t) \end{aligned} \quad (8)$$

このとき、残差 $y_t - f_{N-1}(x_t)$ を弱学習器 h_N で回帰することで、以下のように関数 $f_N(x)$ の更新が行われる。

$$f_N(x) = f_{N-1}(x) + h_N(x) \quad (9)$$

2.4 選択的アンサンブルとは

訓練された弱学習器 $\{h_1, h_2, \dots, h_N\}$ の中から、弱学習器の部分集合を生成することをアンサンブル枝刈り (ensemble pruning) と呼ぶ。枝刈りされたアンサンブル (i.e., 選択的アンサンブル) の汎化性能⁷⁾ は、全ての弱学習器 $\{h_1, h_2, \dots, h_N\}$ を用いたときよりも、改善されることが報告されている (e.g., Zhou et al. 2002)。アンサンブル枝刈りによる利点の一つに、個々の弱学習器の出力を計算する手間を減らし、計算負荷を軽減できることにある。

以下では Hernández-Lobato et al.(2011)で提案された回帰問題に対するアンサンブル枝刈り法について説明する。はじめに、利用可能な N 個の弱学習器 $\{h_1, h_2, \dots, h_N\}$ が存在するとき、単純平均に

よって結合された関数 $f_N(x)$ に対する二乗誤差の期待値は、以下のように示される (Perrone & Cooper 1993)。

$$\begin{aligned} err(f_N) &= \int \left(\frac{1}{N} \sum_{i=1}^N h_i(x) - f^*(x) \right) \left(\frac{1}{N} \sum_{j=1}^N h_j(x) - f^*(x) \right) p(x) dx \\ &= \sum_{i=1}^N \sum_{j=1}^N C_{ij} / N^2 \\ &= N^{-2} \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} C_{ij} + 2N^{-2} \sum_{i=1}^{N-1} C_{iN} + N^{-2} C_{NN} \end{aligned} \quad (10)$$

ただし、

$$C_{ij} = \int (h_i(x) - f^*(x)) (h_j(x) - f^*(x)) p(x) dx \quad (11)$$

Hernández-Lobato et al.(2011)では、弱学習器 $\{h_1, h_2, \dots, h_k\}$ に対する単純平均による誤差 $err(f_k)$ に着目し、式(10)を最小にするような弱学習器の組み合わせを得るために、以下のようなアンサンブル枝刈り法を提案している。

$$s_k = \operatorname{argmin}_k k^{-2} \left(\sum_{i=1}^{k-1} \sum_{j=1}^{k-1} \hat{C}_{s_i s_j} + 2 \sum_{i=1}^{k-1} \hat{C}_{s_i k} + \hat{C}_{kk} \right) \quad (12)$$

ここで、 $k \in \{1, 2, \dots, N\} \setminus \{s_1, s_2, \dots, s_{k-1}\}$,

$$\hat{C}_{ij} = \frac{1}{T} \sum_{t=1}^T (h_i(x_t) - y_t)(h_j(x_t) - y_t) \quad (13)$$

このとき、 $\{s_1, s_2, \dots, s_k\}$ が弱学習器のラベル $\{1, 2, \dots, N\}$ に対する部分集合を表すとすると、訓練誤差が最小となる弱学習器の結合は、以下のように、選択された学習器の単純平均として得られる。

⁷⁾ 検証用データに対する予測性能を指す。

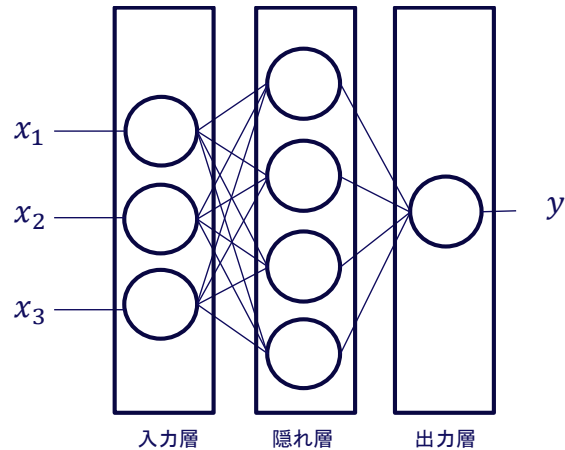
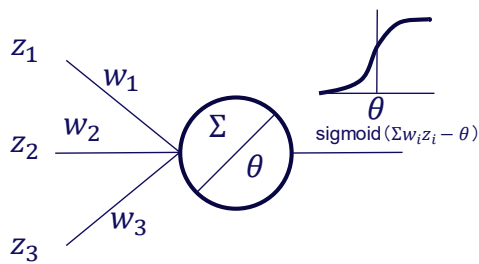


図5 人工素子（ノード）の構造とニューラルネットワークの構造

$$f_k(x) = \frac{1}{k} \sum_{i=1}^k h_{s_i}(x) \quad (14)$$

本研究では、上記に示した Hernández-Lobato et al.(2011)で提案されているアンサンブル枝刈り法を採用する。

3. ニューラルネットワークを用いたアンサンブル学習について

アンサンブル学習では、学習器の集合とそれらを結合した関数を構築する必要がある。このとき、アンサンブル構造を構成する基本学習器 (h) は、通常、決定木やニューラルネットワーク等から選ばれる。アンサンブル学習では、同種の学習器を用いて構成される同種アンサンブルと異なる学習器を用いて構成される異種アンサンブルがある (Zhou 2012)。

本研究では、以下に示す先行研究と同様に、基本学習器としてニューラルネットワークを採用し、同種アンサンブル学習を試みる。

3.1 先行研究

Khwaja et al.(2015)では、米国ニューイングランド地方における電力需要について、予測精度の比較を行っている。この研究では、単一のニューラルネットワークを用いた電力需要予測と比べて、バギングを適用した場合の予測誤差は改善され、予測誤差のばらつきを抑える傾向があることが報告されている。

さらに、Khwaja et al.(2017)では、ブースティングを用いた電力需要予測の精度比較も行っている。彼らは、勾配ブースティングによるアンサンブル学習について考察を行っており、単一のニューラルネットワークやバギングを用いたアンサンブル学習と比べて、勾配ブースティングの方が予測精度の改善がみられ、予測誤差のばらつきが小さくなる傾向があることを示している。

3.2 ニューラルネットワークとは

ニューラルネットワークは、生体の神経細胞（ニューロン）を模擬した人工の素子（ノード）を相互に接続して構成される（図5参照）。ニューラルネットワークに使われる人工の素子は、生

体のニューロンを厳密に模倣したものではなく、その特定の機能を抽出し、単純化した数理モデルである。

人工の素子では、情報が与えられると、事前に与えられた閾値 θ を超えたときに活性化し、ネットワーク構造に従って、他の素子へと情報が伝達される（図 5）。この一連の動きを表現した関数を活性化関数と呼ぶ。例えば、素子 j に関する活性化関数を以下のようなシグモイド関数で表す。

$$\text{sigmoid}(\alpha_j) = \frac{1}{1 + \exp(-\alpha_j)} \quad (15)$$

$$\alpha_j = \sum_i w_{ji} z_i - \theta_j \quad (16)$$

ここで、 w_{ji} は素子 i の出力値 z_i の重み（パラメータ）を表す。

図 5 より、ニューラルネットワークでは、入力データ x が与えられると、入力層から出力層に向かって、情報の伝達が素子を通じて行われた後、誤差逆伝搬法により、各素子に含まれる重み w と閾値 θ が更新されることで、学習が行われる。

3.3 誤差逆伝搬法を用いたパラメータ推定

ニューラルネットワークは、重み w と閾値 θ の値を変えることにより、入出力関係を変化させる。誤差逆伝搬法では、勾配法に基づいて、ネットワーク内部のパラメータを更新する。

勾配法では、ネットワーク内部のパラメータに適切な初期値を与えた上で、観測値 y_t と予測値 \hat{y}_t

に関する誤差尺度 E を小さくするように、勾配（偏微分）を用いてパラメータの微修正を繰り返す。この手続きを繰り返すことで、パラメータの値は、極小値の 1 つに収束する。例えば、重み w の更新は以下のように行われる。

$$w^{new} \leftarrow w^{old} - \eta \sum_{t \in B} \frac{\partial E(y_t, \hat{y}_t)}{\partial w} \quad (17)$$

ここで $\eta (> 0)$ は学習率パラメータを表す。また、 B は各ステップで訓練用データからランダムに選んだデータ集合（i.e., ミニバッチ）を表している。ミニバッチサイズが 1 の場合を、確率的勾配降下法と呼ぶ。

式 (17) における勾配の評価には、連鎖律⁸⁾ を利用する。素子 i から素子 j への重みを w_{ji} とし、素子 j の活性化関数（e.g., シグモイド関数）を $z_j = \sigma(\alpha_j)$ 、誤差尺度 $E_t = E(y_t, \hat{y}_t)$ とおく。このとき、出力層の素子数が 1 つであれば、勾配は以下のように表される。

$$\frac{\partial E_t}{\partial w_{ji}} = \frac{\partial E_t}{\partial \alpha_j} \frac{\partial \alpha_j}{\partial w_{ji}} = \frac{\partial E_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_j} \frac{\partial z_j}{\partial \alpha_j} \frac{\partial \alpha_j}{\partial w_{ji}} \quad (18)$$

ここで、例えば、出力層における観測値 y_t の予測値 \hat{y}_t を線形回帰モデル

$$\hat{y}_t = \sum_j w_{1j} z_j - \theta_1 \quad (19)$$

で表し、誤差尺度 E_t を以下のような二乗誤差とすれば、

⁸⁾ 微分法における連鎖律は、合成関数 $f(g(x))$ を x で微分するとき、以下のように表される。

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f}{\partial g} \frac{\partial g}{\partial x}$$

表 1 気象要因と電力需要の基本統計量（東京エリア、期間 2016/4/1～2019/3/31）

	気圧(hPa)		降水量(mm)	気温(°C)	露点温度(°C)	蒸気圧(hPa)	湿度(%)
	現地	海面					
平均	1010.7	1013.5	0.9	17.2	11.1	15.8	70.1
分散	52.6	53.4	4.8	66.9	101.9	79.3	396.0
最大値	1033.0	1036.1	40.0	37.4	28.4	38.7	100.0
最小値	965.5	968.2	0.0	-3.6	-17.2	1.6	16.0
	風速(m/s)	日照時間(h)	全天日射量(MJ/m ²)	降雪(cm)	積雪(cm)	電力需要(MW)	
平均	2.9	0.4	1.0	2.1	2.7	32736	
分散	2.2	0.2	0.9	14.3	12.6	41661383	
最大値	15.3	1.0	4.0	13.0	9.0	56532	
最小値	0.0	0.0	0.0	0.0	0.0	19768	

・電力需要は東京エリア（栃木県、群馬県、茨城県、埼玉県、千葉県、東京都、神奈川県、山梨県、静岡県の一部）、気象データは東京都のみ

$$E_t = \frac{1}{2} (y_t - \hat{y}_t)^2 \quad (20)$$

式(18)の勾配を容易に求めることができる。

4. 電力需要予測の精度比較

4.1 データの概要

本研究では、電力広域的運営推進機関で公表されている 1 時間毎の電力需要データ（MW、東京エリア）を用いる。分析期間は 2016/4/1～2019/3/31 で、訓練用データとして 2016/4/1～2018/3/31 までの 2 年間を用いて、パラメータの推定を行う。その後、2018/4/1～2019/3/31 までの電力需要を予測し、学習方法の違いによる予測精度の違いについて、月別、曜日別、時間別に比較を行う。

また、入力データとして、気象庁が公表している 1 時間毎の気象データ（東京）を用いる。気象データには、気圧（現地、海面）、降水量、気温、露点温度、蒸気圧、湿度、風速、日照時間、全天

日射量、降雪、積雪が含まれる（表 1）。さらに年データ（数値）、月ダミー、日ダミー、曜日ダミー、時間ダミーを加えた。分析に用いるデータは、最小値 0、最大値 1 となるように基準化している。

4.2 ニューラルネットワークの設定

アンサンブル学習に用いるニューラルネットワークは、先行研究（Khwaja et al. 2015, 2017）と同様に隠れ層（中間層）の数を 1 層とした 3 層構造（入力層、隠れ層、出力層）とする。また、隠れ層の素子数（ノード数）は、事前に交差検証法（Cross-Validation）⁹⁾ によって決定する。つまり、アンサンブルに含まれるニューラルネットワークの隠れ層と素子の数は、全ての学習器で同じである。

ニューラルネットワークのパラメータは、勾配法に基づく誤差逆伝搬法によって、推定を行う。はじめに、パラメータの初期値が与えられると、勾配（偏微分）が計算され、パラメータの更新が行われる。本分析では学習器ごとに異なる初期値

⁹⁾ 素子数の決定は 10 分割交差検証を用いて、最大 10 個まで検証を行い、汎化性能が高いものを採用した。

表 2 予測誤差の比較（東京エリア、期間 2018/4/1～2019/3/31）

学習率 パラメータ		バギング			ブースティング		
		Single	Bagging	Bagging.SE	AdaBoost.R	AdaBoost.R2	GdBoost
0.25	RMSE	1967	1910	1909	1818	1918	1938
	増減率		-2.9%	-2.9%	-7.6%	-2.5%	-1.5%
	学習器の数		7	5	9	8	3
0.50	RMSE	1939	1848	1743	1730	1687	1901
	増減率		-4.7%	-10.1%	-10.8%	-13.0%	-2.0%
	学習器の数		10	2	5	2	7
0.75	RMSE	2070	2004	1991	1931	1993	1995
	増減率		-3.2%	-3.8%	-6.7%	-3.7%	-3.6%
	学習器の数		7	2	3	2	6

- ・東京エリアは、栃木県、群馬県、茨城県、埼玉県、千葉県、東京都、神奈川県、山梨県、静岡県の一部。
- ・増減率は、単一学習器（Single）に対する RMSE（Root Mean Squared Error）の変化を表す。
- ・ニューラルネットワークの訓練回数は 1000 回、勾配法の学習率パラメータ（ η ）は 3 パターン {0.25, 0.50, 0.75} を採用。
- ・アンサンブル学習における弱学習器の数は RMSE が最小になるものを採用、ただし最大 10 個まで $\{h_1, h_2, \dots, h_{10}\}$ 。
- ・Bagging. SE は、Bagging に選択的アンサンブルを適用したケースを指す。

を与え、パラメータの更新を 1000 回繰り返すことで、予測を行うためのパラメータを推定する。

4.3 予測精度の比較結果

本研究では、学習方法の違いにおける予測精度の比較を行う。比較する学習方法は、バギング（Bagging）、バギングと選択的アンサンブルを組み合わせた方法（Bagging.SE）、AdaBoost.R、AdaBoost.R2、勾配ブースティング（GdBoost）の 5 種類のアロリズムを用いる。また、学習パラメータの設定によって、学習結果が変わる可能性があるため勾配法の学習率パラメータ η ($0 < \eta \leq 1$) は、 $\eta = \{0.25, 0.50, 0.75\}$ の 3 ケースについて計算を試みた。

表 2 では、各学習方法の RMSE（Root Mean Squared Error）と単一学習器（Single）に対する RMSE の増減率、さらにアンサンブル学習に用いた弱学習器の数を示している。

表 2 をみると、RMSE の増減率が負値であることから、単一学習器と比較すると、アンサンブル学習を用いることで、予測誤差が改善しているこ

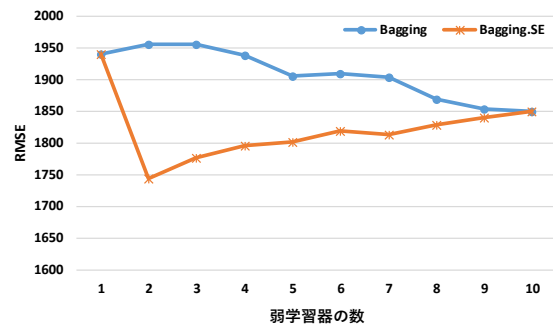


図 6 Bagging と Bagging. SE の予測誤差

（学習率パラメータ=0.50、期間 2018/4/1～2019/3/31）

とが分かる。ただし、学習方法や学習率パラメータの設定などで、RMSE の増減率に違いがみられる。

また、バギングと選択的アンサンブルを組み合わせること（Bagging.SE）で、通常バギングよりも少ない弱学習器で RMSE が小さくなる傾向がみられる（表 2、図 6）。特に、学習率パラメータが 0.50 であるとき、弱学習器の数が大幅に減少している。

予測誤差の分布（付図 A.1）をみると、0 を中心として、左に裾の長い形状をしていることから、観測値よりも予測値の方が大きく、過大評価して

いる傾向がみられる。

月別に予測誤差の分布（付図 A.2）を確認すると、単一学習器では、1月の予測が大きく外れる傾向がみられるが、単一学習器と比べて、アンサンブル学習の予測誤差は、ばらつきが小さく抑えられ、1月の予測誤差が改善しているケースがみられる。

次に、曜日別の予測誤差の分布を付図 A.3 に示す。付図 A.3 より、火曜日の予測誤差は、他の曜日よりも下方に分布している傾向がみられる。しかし、アンサンブル学習では、単一学習器と比較して、誤差のばらつきを小さく抑えられているケースがみられ、火曜日の予測誤差が改善されている。

最後に、時間別に予測誤差の分布（付図 A.4）をみると、正午前後で予測誤差が大きくなる傾向がみられる。特に、単一学習器の予測誤差は大きく下方に分布している傾向がみられる。一方で、アンサンブル学習の予測誤差のばらつきは、単一学習器と比べて小さくなる傾向がみられる。

5. 考察とまとめ

本研究では、アンサンブル学習を電力需要予測に適用し、予測精度を比較することで、アンサンブル学習の有効性について検証を行った。その結果、アンサンブル学習を適用することで、予測誤差（RMSE）の改善がみられた。

また、月別、曜日別、時間別にアンサンブル学習の予測結果を確認すると、予測誤差のばらつきが小さくなり、予測精度が改善されている傾向がみられた。特に、単一学習器の予測誤差が大きくなる時点で、改善傾向が顕著にみられる。このことから、単一学習器では予測が難しい時点でも、複数の学習器を組み合わせることで、予測誤差が改善する可能性が考えられる。

最後に、今回の分析ではアンサンブル学習を適用することで、予測誤差の改善がみられたが、予

測精度を上げるために、多くの弱学習器が必要となる場合には、単一学習器よりも計算負荷が大きくなる点に留意する必要がある。さらに、アンサンブル学習では複数のモデルを集約するため、モデルの解釈性を欠くことになる。そのため、気象要因と電力需要の関係は、ブラックボックスであることに注意が必要である。

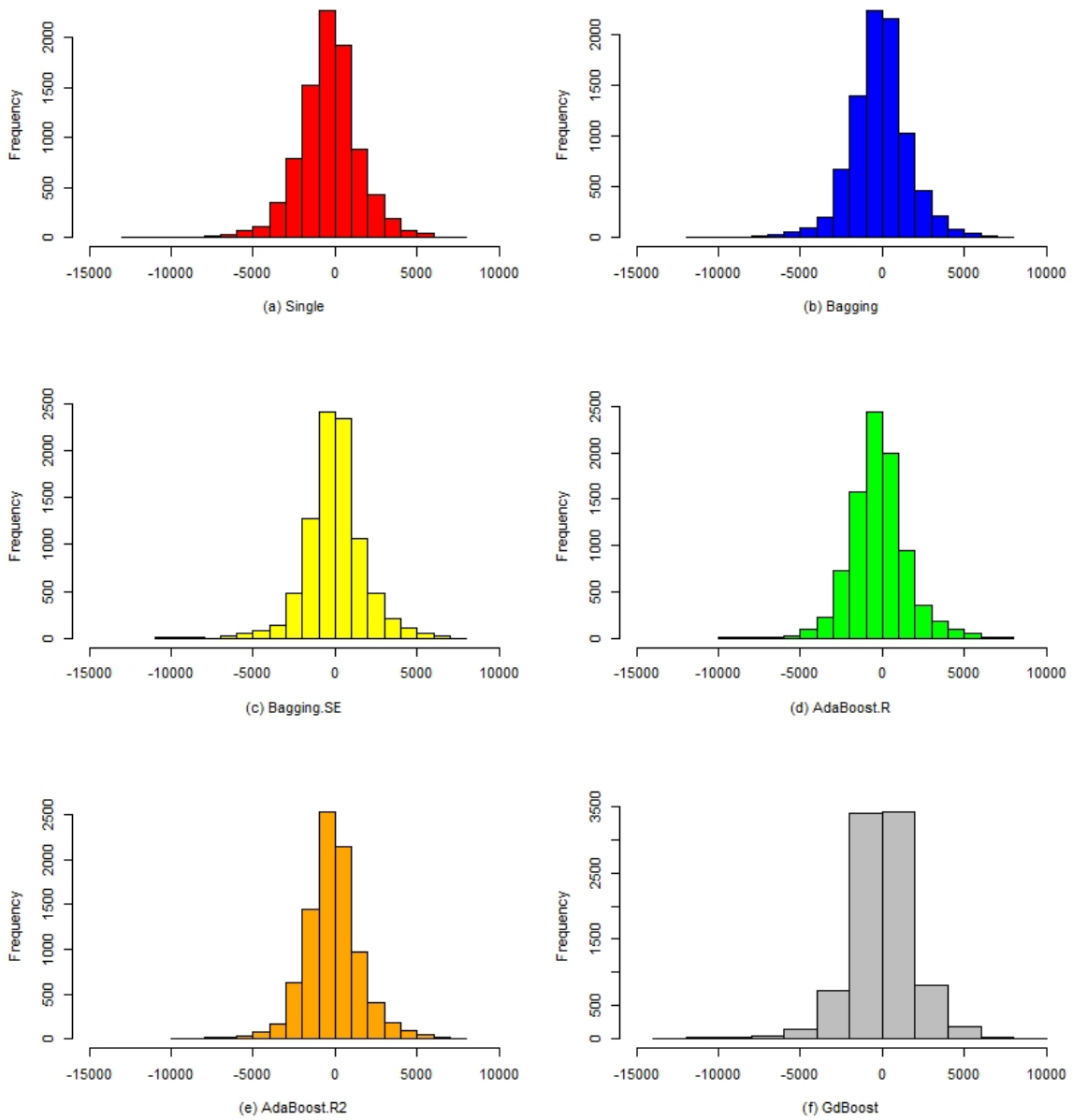
参考文献

- [1] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2), 123-140.
- [2] Drucker, H. (1997, July). Improving regressors using boosting techniques. In *ICML (Vol. 97, pp. 107-115)*.
- [3] Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In *International workshop on multiple classifier systems (pp. 1-15)*. Springer, Berlin, Heidelberg.
- [4] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1), 119-139.
- [5] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.
- [6] Hernández-Lobato, D., Martínez-Muñoz, G., & Suárez, A. (2011). Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles. *Neurocomputing*, 74(12-13), 2250-2264.
- [7] Khwaja, A. S., Naeem, M., Anpalagan, A., Venetsanopoulos, A., & Venkatesh, B. (2015). Improved short-term load forecasting using bagged neural networks. *Electric Power Systems Research*, 125, 109-115.
- [8] Khwaja, A. S., Zhang, X., Anpalagan, A., & Venkatesh, B. (2017). Boosted neural networks for

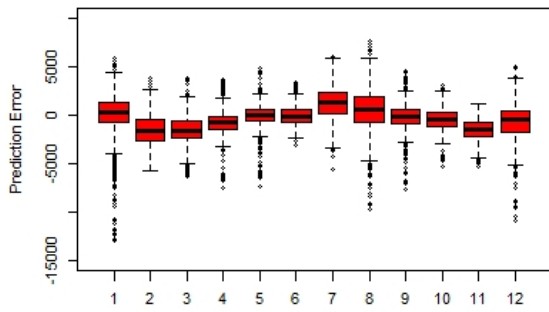
improved short-term electric load forecasting.
Electric Power Systems Research, 143, 431-437.

- [9] Perrone, M. P., & Cooper, L. N. (1993). When Neural Networks Disagree: Ensemble Methods for Hybrid Neural Networks. In RJ Mammone (Ed.), Artificial Neural Networks for Speech and Vision. London, UK: Chapman & Hall.
- [10] Zhou, Z. H., Wu, J., & Tang, W. (2002). Ensembling neural networks: many could be better than all. Artificial intelligence, 137(1-2), 239-263.
- [11] Zhou, Z. H. (2012). Ensemble methods: foundations and algorithms. Chapman and Hall/CRC.

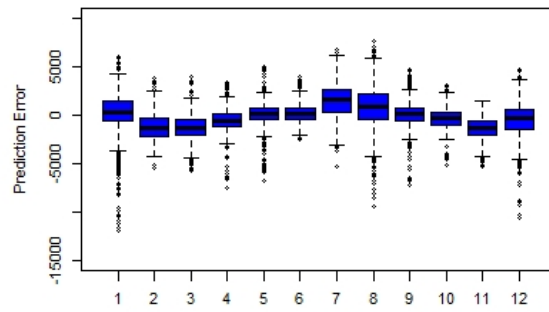
付録A 予測誤差の分布



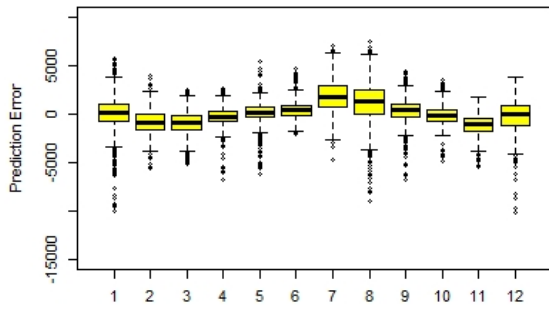
付図 A.1 予測誤差の分布 (学習率パラメータ=0.50、期間 2018/4/1~2019/3/31)



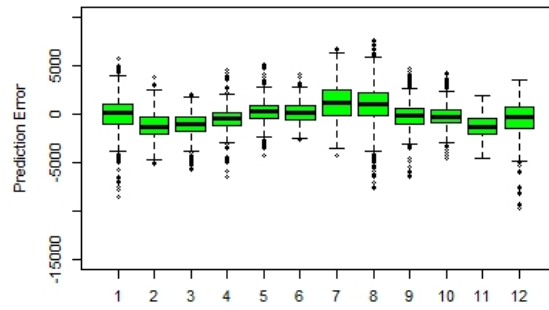
(a) Single



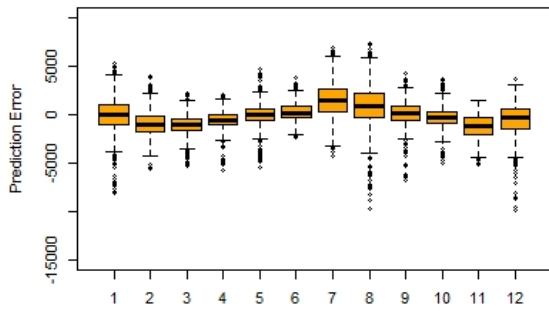
(b) Bagging



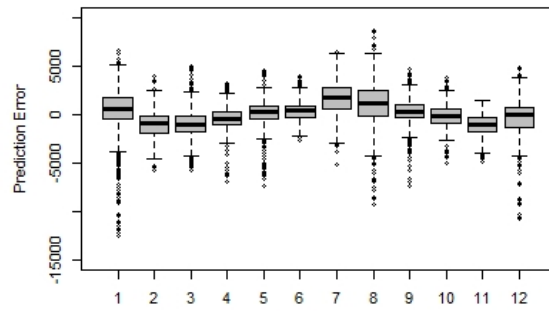
(c) Bagging.SE



(d) AdaBoost.R

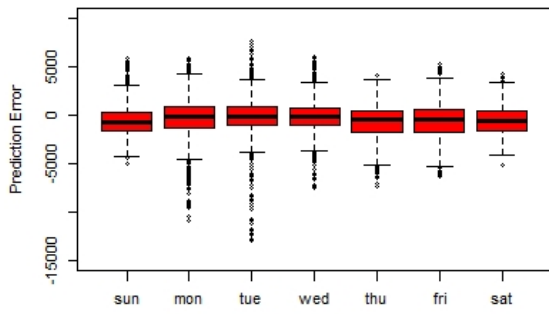


(e) AdaBoost.R2

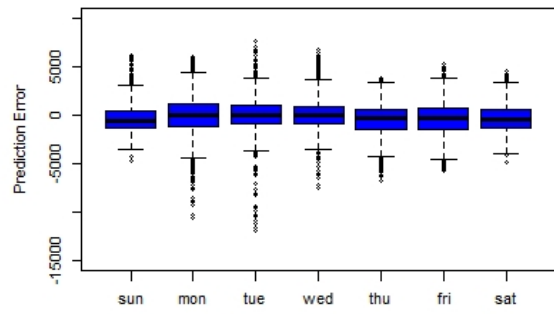


(f) GdBoost

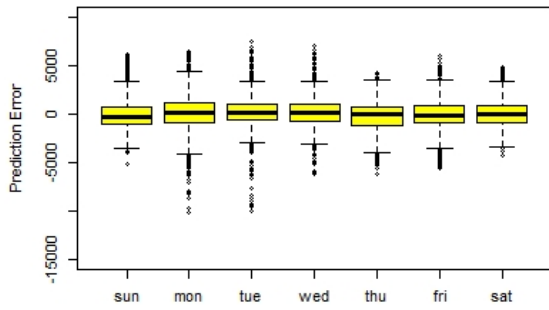
付図 A. 2 月別予測誤差の分布 (学習率パラメータ=0.50、期間 2018/4/1~2019/3/31)



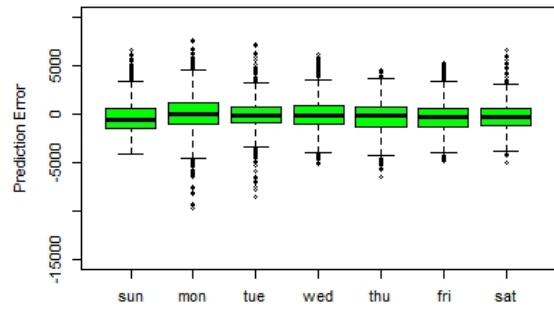
(a) Single



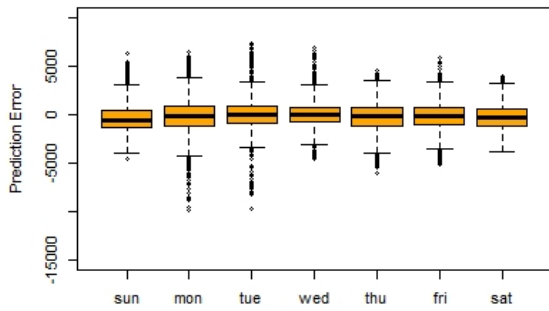
(b) Bagging



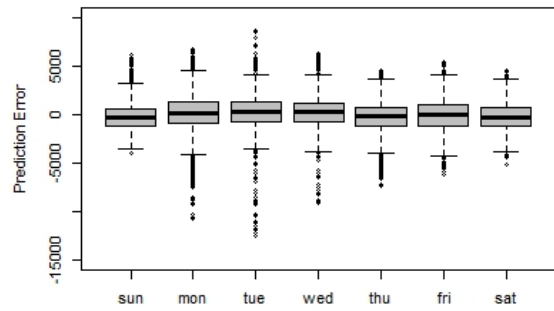
(c) Bagging.SE



(d) AdaBoost.R

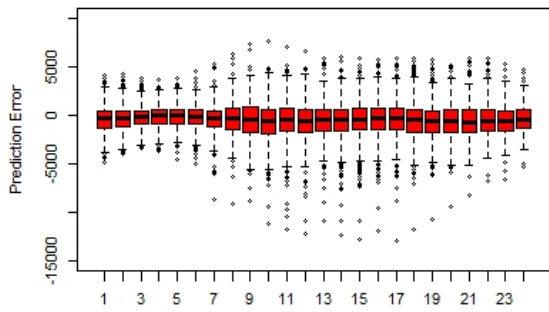


(e) AdaBoost.R2

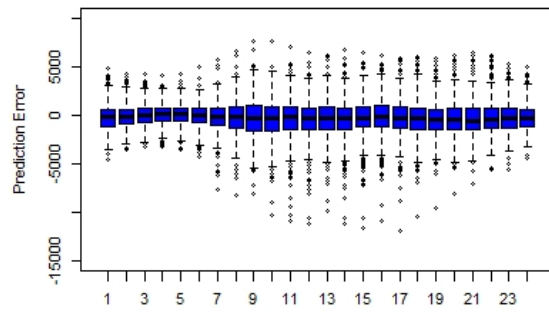


(f) GdBoost

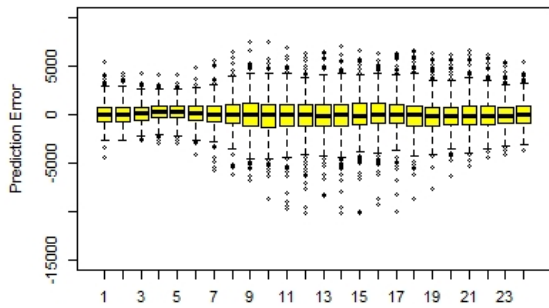
付図 A.3 曜日別予測誤差の分布 (学習率パラメータ=0.50、期間 2018/4/1~2019/3/31)



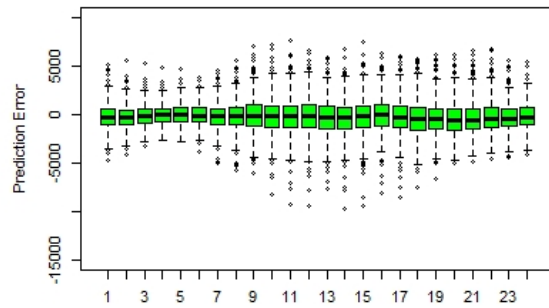
(a) Single



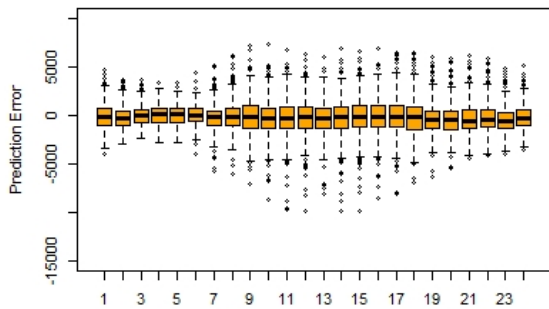
(b) Bagging



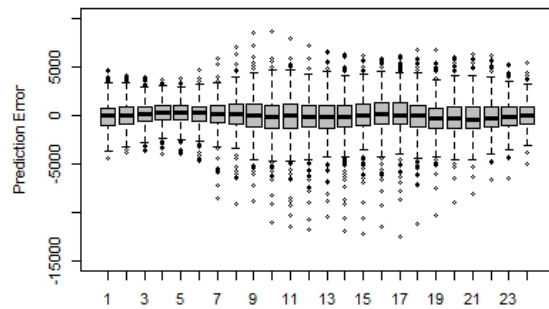
(c) Bagging.SE



(d) AdaBoost.R



(e) AdaBoost.R2



(f) GdBoost

付図 A. 4 時間別予測誤差の分布 (学習率パラメータ=0.50、期間 2018/4/1~2019/3/31)

IR

CRIEPI

Central Research Institute of
Electric Power Industry

[不許複製]

編集・発行人 一般財団法人 電力中央研究所
社会経済研究所長
東京都千代田区大手町1-6-1
e-mail src-rr-ml@criepi.denken.or.jp

発行・著作・公開 一般財団法人 電力中央研究所
東京都千代田区大手町1-6-1
