Vol.2017-DBS-166 No.17
2017/12/23

モデル圧縮における擬似データ生成手法の提案

河野 晋策[1,a)]　若林 啓[2,b)]

概要：機械学習による分類において，精度の高い手法として複数の分類モデルの結合であるアンサンブルがよく用いられるが，アンサンブルは多大な計算資源を必要とするため，携帯端末など計算資源の限られた環境で用いるのが難しい．この問題に対して，アンサンブルを小さなニューラルネットワークで近似するモデル圧縮の手法が提案されている．モデル圧縮では，オリジナルデータを基に大量の擬似データを生成して近似モデルの学習に用いるが，この擬似データ生成において真のデータ分布をよく近似した擬似データ分布を得ることが，近似モデルの性能を元のアンサンブルに近づけるために重要である．本研究では，分類クラスごとの分布の偏りを考慮することで，既存手法よりも近似モデルの学習に有効な擬似データを高速に生成する Adaptive MUNGE を提案する．実験により，提案手法は既存手法と比較して高速に擬似データを生成することができ，かつ，より精度を保つモデル圧縮が実現できることを示す．

1. はじめに

近年，機械学習の分野で，分類器のアンサンブルを使用することで，単一のモデルよりも良いパフォーマンスを発揮することがよく知られている [1], [2], [3]．アンサンブルはその予測が加重平均あるいは投票によって結合されたモデルの集合であり，様々なアンサンブル手法が提案されている [1], [4]．しかし，多くのアンサンブルは構造が大きく複雑なため，訓練や実行に時間がかかるという欠点がある [5]．このため，メモリやストレージスペースが限られたデバイスやアプリケーションでは，アンサンブル手法が使用できない場合がある．例えば，携帯端末には厳しいメモリとストレージの制限があり，モデルを数メガバイトのパラメータに制限しなければならないことがある．また，リアルタイムの予測が必要なアプリケーションなど，高速に分類を行う必要のある場面では，アンサンブルの使用を諦めて軽量な分類器を使わなければならないこともある．

このような状況に対して，任意の関数を近似可能であるというニューラルネットワークの特徴を利用して，精度を下げることなく，高速かつコンパクトなモデルを得るモデル圧縮の手法が提案されている [6], [7], [8]．モデル圧縮は，以下の手順を踏むことで高精度なアンサンブルを単一モデルによって近似する．

(1) 訓練データに対して，アンサンブルを学習
(2) 訓練データを元に擬似データを生成
(3) 1 で訓練したアンサンブルを用いて 2 で生成した擬似データにクラスラベルを付与
(4) 3 のクラスラベル付き擬似データを用いて，単一モデルを訓練

アンサンブルによってクラスラベルを付与された擬似データを単一モデルで学習することで，元のアンサンブルモデルの出力を単一モデルによって近似し，元の訓練データでは実現できない精度の単一モデルを訓練することが可能となる．結果として，アンサンブルの精度を保ち，アンサンブルよりも高速かつコンパクトなモデルを得ることができる．モデル圧縮の基本的な考え方は，初めに与えられた訓練データを遥かに上回る大量のデータを用いてモデルを訓練することで，元の訓練データで実現できないパフォーマンスを達成することである．しかし，大量のデータを追加で取得することは多くの場合には難しい．このため，元の訓練データから擬似データを生成する 2 のステップがモデル圧縮において重要な位置付けになっている．

擬似データの生成手法としては，MUNGE[7]，Model Based Sampling[8] が提案されている．MUNGE は，各訓練データに対して最近傍を発見し，その値を交換することで擬似データを生成する．しかし，分類タスクにおいてクラスの境界にデータを生成してしまい，モデルの学習の妨げとなる可能性がある．一方，Model Based Sampling は，アンサンブルの候補となる決定木の決定パスを用いて擬似データを生成することで，データの分布により忠実な擬似データ生成を実現する．この手法によって MUNGE と比較して優れたモデル圧縮を行えることが実験的に示されて

[1] 筑波大学 情報学群 知識情報・図書館学類
[2] 筑波大学 図書館情報メディア系
[a)] s1411524@klis.tsukuba.ac.jp
[b)] kwakaba@slis.tsukuba.ac.jp

いるが，Model Based Sampling は決定木のみをアンサンブルの候補とした場合にしか適用できない．

本研究では，モデル圧縮で必要となる擬似データ生成において，どのようなアンサンブルにも適用可能な Adaptive MUNGE を提案する．提案手法は分類タスクに適用することを想定した手法であり，分類クラス毎に擬似データを生成することで，擬似データの多様性を保つ．これにより，外れ値によって各クラスが入り混じった状態である場合に，クラスの境界面が明確になり，元の訓練データで訓練したニューラルネットよりもモデル圧縮により得られたモデルがより高精度に分類可能であることを示す．また，少数派クラスのデータを増やし，クラスごとのデータ数の偏りを無くすことで，不均衡データに対して既存研究よりも提案手法の方が良いパフォーマンスを発揮することを示す．

本稿の構成は以下の通りである．2 章で本研究と関連するモデル圧縮，及び，アンサンブルに関する研究について概観し，本提案手法の妥当性について議論する．3 章で先行研究の課題とその解決策を述べ，提案手法の Adaptive MUNGE のアルゴリズムを示す．4 章で提案法と既存手法の比較実験を行った上で，提案手法の有用性を明らかにする．5 章で本稿のまとめと今後の展望について述べる．

2. 関連研究

2.1 モデル圧縮

近年，モデル圧縮に関する研究は，盛んに行われている [6], [7], [8]．Zeng と Martinez [6] は，ニューラルネットを用いたアンサンブルの近似手法を提案した．彼らは，単一の隠れ層を持った 10 個のニューラルネットを結合したものをアンサンブルとし，訓練データの周辺分布からランダムに取得したデータを擬似データとして用いている．しかし，Zeng と Martinez は，擬似データを用いても，元の訓練データによって訓練したニューラルネットよりも大きな改善がないと結論づけた．これに対して，Buciluǎ ら [7] は，より複雑なアンサンブルを圧縮するためには，より多くの擬似データが必要であることを示している．Buciluǎ らは，Zeng と Martinez のアンサンブルはモデル圧縮が必要なほど複雑なモデルではなかったことを指摘した上で，複雑なアンサンブルのモデル圧縮を行うためには，より多くの擬似データが必要であることを示した．この擬似データを生成するために，Buciluǎ は，擬似データ生成手法である MUNGE を提案した．MUNGE は入力データに対して，ユークリッド空間における最近傍を見つけ，確率パラメータ p と分散パラメータ s に基づいて，正規分布から新規データを生成する．より詳細には，訓練データのインスタンス e とその最近傍 e' の属性 e_a，e'_a について，連続属性の場合，標準偏差 $|e_a - e'_a|/s$，平均 e'_a とする正規分布から新しい値 e_a が生成される．非連続属性の場合，e_a と e'_a の値が交換される．MUNGE のアルゴリズムを Algorithm 1 に示す．

Algorithm 1 MUNGE

Require:
訓練データセット（クラスラベルなし）T, イテレーション回数 k, 確率パラメータ p, 分散パラメータ s
$Norm(a, b)$:平均 a, 標準偏差 b の正規分布から引いたランダム値
Ensure: $k \times size(T)$ のクラスラベルなしデータセット D

$D \leftarrow \phi$
loop k 回:
  $T' \leftarrow T$
  for all T' のインスタンス e **do**
    $e' \leftarrow T'$ 内の最近傍 e
    for all e の特徴量 a **do**
      p の確率で:
      if a が連続属性の場合 **then**
        $sd \leftarrow |e_a - e'_a|/s$
        $e_a \leftarrow Norm(e'_a, sd)$, $e'_a \leftarrow Norm(e_a, sd)$
      else
        e の特徴量と e' の特徴量を入れ替える
      end if
    end for
  end for
  $D \leftarrow D \cup T'$
end loop

しかし，MUNGE ではクラスの境界にデータを生成し，モデルの学習の妨げとなる可能性がある．Lindgren [8] は，モデル圧縮における擬似データ生成手法として Model Based Sampling を提案し，MUNGE に比べて優れたモデル圧縮が実現できることを示した．Model Based Sampling はアンサンブルの候補として決定木のみを用いる．決定木のパスを活用することで，データの表現に多様性を持たせ，元の訓練データで訓練可能な決定木よりも高精度な決定木の訓練を行うことができる．しかし，この手法は決定木のみを候補とするアンサンブルにおいてのみ有効であり，決定木がうまく当てはまらないデータに対しては活用できない．本研究では，任意のアンサンブルにおいてもモデル圧縮を適用可能な擬似データ生成手法を提案する．

モデル圧縮における擬似データ生成は，不均衡データ学習におけるオーバーサンプリングと密接に関連する．不均衡データに対して，少数派クラスをオーバーサンプリングすることで，モデルのパフォーマンスが向上することが示されている [9], [10], [11]．Liu ら [9] は不均衡データ学習の場面で事前分布に基づいて生成的に少数派のクラスデータをオーバーサンプリングする手法を提案した．本研究では，このアイデアに基づいて，不均衡データに対して先行研究よりも圧縮後のモデルのパフォーマンスを向上する手法を提案する．

2.2 アンサンブル

アンサンブルとは，その予測が加重平均あるいは投票によって結合されたモデルの集合である [4]．しかし，多くのアンサンブルはその構造が大きく複雑なため，訓練や実

行に時間がかかる [5]．Buciluǎ ら [7] は，アンサンブル構築にアンサンブル選択 [12] を用い，多くの擬似データを用いることで複雑なアンサンブルでも圧縮可能なことを示した．アンサンブル選択では，既に訓練された候補モデルであっても，アンサンブルのパフォーマンス向上に繋がらないモデルは，排除される．このように候補モデルの全てを結合するのではなく，部分集合を選択することはアンサンブル枝刈りと呼ばれ，より小さなサイズのアンサンブルでより良い汎化性能を得ることが期待される [12], [13], [14]．Tsoumakas ら [15] は，アンサンブル枝刈りを順序付けに基づく枝刈り，クラスタリングに基づく枝刈り，最適化に基づく枝刈りの 3 つのカテゴリーに分類した．また，Hernández-Lobato ら [16] は，最適化に基づく枝刈りと順序付けに基づく枝刈りは，一般に，Adaboost.R2 アルゴリズム，Negative Correlation Learning または Regularized Linear Stacked Generalization によって生成された他のアンサンブル，及び，他のアンサンブル枝刈りによって得られたアンサンブルモデルよりも優れていることを報告している．本研究では，計算時間を鑑み，最適化に基づく枝刈りを使用し，アンサンブルの結合と枝刈りを行う．

最適化に基づく枝刈りでは，アンサンブル枝刈りの問題をアンサンブルの一般化性能に関係する目的関数について最大化（最小化）するパラメータを探すことを目的とした最適化問題へと帰着させる．

Zhou ら [13] は，アンサンブルの重み付け結合における理論的最適解は現実的に導出不可能であるとし，アンサンブル枝刈り問題を最適化問題としてみることで，GASEN を提案した．GASEN は，各モデルに対する重みベクトルの集合をランダムに設定し，遺伝的アルゴリズムによって，テストデータに対する各重みベクトルの適合度を計算する．もっとも最適な重みベクトルに基づいて，アンサンブルを構築し，重みが小さいモデルは除外する．

Li と Zhou [14] はアンサンブル枝刈りを効率的に解くことができる QP 問題へと帰着させる RSE（regularized selective ensemble）アルゴリズムを提案した．RSE は，スパース誘導性を持つ l_1 ノルム制約を導入することで，自然に枝刈りを行い，先行の枝刈りよりも小さいサイズで汎化能力の高いアンサンブルを生成する．

M 個のモデル $\{h_1,\ldots,h_M\}$ に対し，アンサンブル結合重みベクトルを $\boldsymbol{w} = [w_1,\ldots,w_M]^{\mathrm{T}}$ と定義する．この時，$w_i \geq 0$ かつ $\sum_{i=1}^{M} w_i = 1$ である．RSE は，正則化リスク関数 $R(\boldsymbol{w}) = \lambda V(\boldsymbol{w}) + \Omega(\boldsymbol{w})$ を最小化することにより $\boldsymbol{w}$ を決定する．ここで，$V(\boldsymbol{w})$ は訓練データ $D = \{(\boldsymbol{x}_1, y_1),\ldots,(\boldsymbol{x}_N, y_N)\}$ に対する誤分類の経験損失で，$\Omega(\boldsymbol{w})$ は正則化項であり，λ は $V(\boldsymbol{w})$ と $\Omega(\boldsymbol{w})$ の最小化における正則化パラメータを表す．ヒンジ損失とグラフラプラシアン正則化項をそれぞれ経験損失と正則化として用いることにより，問題は下式（1）で定式化される．

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \boldsymbol{w}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{L}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{w} + \lambda\sum_{i=1}^{N}\max(0, 1 - y_i\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{w}) \\ \text{s.t.} \quad & \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1,\ \boldsymbol{w} \geq \mathbf{0}. \end{aligned} \tag{1}$$

ここで，$\boldsymbol{p}_i = (h_1(\boldsymbol{x}_i),\ldots,h_M(\boldsymbol{x}_i))^{\mathrm{T}}$ は訓練データ $\boldsymbol{x}_i$ に対する個々のモデルの予測を表し，$\boldsymbol{P} \in \{-1,+1\}^{M\times N}$ は全訓練データに対する全モデルの予測を集めた予測行列で，$\boldsymbol{P}_{ij} = h_i(\boldsymbol{x}_j)$ である．$\boldsymbol{L}$ は訓練データの近傍グラフ G の正規化グラフラプラシアンである．式（1）の $max(\cdot)$ は滑らかではないので，スラック変数 $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_m)^{\mathrm{T}}$ を導入することにより，式（1）は下式（2）で書き表せる．

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \boldsymbol{w}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{L}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{w} + \lambda\mathbf{1}^{\mathrm{T}}\boldsymbol{\xi} \\ \text{s.t.} \quad & y_i\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{w} + \xi_i \leq 1,\ (\forall i = 1,\ldots,N) \\ & \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1,\ \boldsymbol{w} \geq \mathbf{0},\ \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned} \tag{2}$$

この時，式（2）は標準的な QP 問題となり，従来の最適化パッケージを用いて，効率的に解くことができる．また，$\mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1, \boldsymbol{w} \geq \mathbf{0}$ という制約は、スパース誘導性を持つ l_1 ノルム制約となり，重み $\boldsymbol{w}$ のいくつかの要素を強制的にゼロにする．導出された結合重みベクトル $\boldsymbol{w}$ を用いて，式（3）のように $\boldsymbol{w}$ の要素がゼロでない候補モデルの投票により予測を決定する．また，下式（4）のように重み結合アンサンブルも提案されている．

$$H(\boldsymbol{x}) = \sum_{w_i>0} h_i(\boldsymbol{x}) \qquad \text{(RSE)} \tag{3}$$

$$H(\boldsymbol{x}) = \sum_{w_i>0} w_i h_i(\boldsymbol{x}) \quad \text{(RSE-w)} \tag{4}$$

3. モデル圧縮における擬似データ生成

擬似データ生成について問題となるのは，擬似データの分布を実際の訓練データの分布によく一致させることである．例えば，正規分布を用いて擬似データの生成を行う場合，真の分布が別の分布である場合，一部のデータに対してうまく表現できない．実際，データを生成するアルゴリズムには分散やバイアスが存在し，真の多様体を任意の仮説で表現することは非常に難しい．しかし，仮説の結合により，これらの分散やバイアスが減少することがある [2], [3]．Model Based Sampling[8] は，擬似データ生成に決定木の決定パスを用いることで，多数の正規分布からデータを表現し，決定木における圧縮で MUNGE[7] よりも有意に性能が向上することを示した．しかし，Model Based Sampling はアンサンブルが決定木で構成されている必要があり，任意のアンサンブルを圧縮することはできない．任意のアンサンブルに対して圧縮が可能な MUNGE は，パラメータに対する経験的知見は示されておらず，また，訓練データのクラス分布に偏りが存在する場合，擬似データ生成により，さらにクラスの偏りが顕著になり，また，クラスの境

界線を鑑みないため，クラスの境界に当たる擬似データを生成し，モデルの学習の妨げになることがある．

Model Based Sampling のように，データセットをある単位に分割し，その単位で生成分布が仮定できる場合，データセット全体に対して仮説を考えるよりも，アルゴリズムの分散やバイアスを軽減できる可能性がある．提案手法では，データセットをクラスラベルによって分割し，クラスラベル毎に擬似データを生成することで擬似データ生成アルゴリズムの分散やバイアスを軽減することを目指す．また，クラスラベル毎に擬似データを生成することでデータの不均衡を改善し，不均衡データに対する圧縮をより効率的に行うことを目指す．

3.1 Adaptive MUNGE

本研究では，MUNGE よりもパラメータを減少させ，さらに，真の分布をよりうまく近似する擬似データを生成する Adaptive MUNGE を提案する．MUNGE との大きな違いは，クラスラベルに基づき，擬似データを生成する点である．これは，各インスタンスのクラスとその最近傍として挙げられるインスタンスのクラスを一致させることによって実現する．与えられた訓練データセットのクラス T_c について，各インスタンス e についてユークリッド距離に基づき，最近傍 e' を発見する．この時，離散属性はワンホットエンコーディングされ，連続属性は標準化されている．特徴量が連続属性の場合，式 (5) を用いて，擬似データの値をサンプルする．

$$e_a \leftarrow e_a - u|e_a - e'_a| \quad u \sim U(0,1) \tag{5}$$

式 (5) を用いることでインスタンス e とその最近傍 e'_a の直線間に来るように擬似データの値を得ることができ，さらに MUNGE に存在した分散パラメータ s を廃止することができる．e'_a についても同様にサンプルする．特徴量が非連続属性の場合，MUNGE と同様に e_a と e'_a の値を交換する．Adaptive MUNGE のアルゴリズムを Algorithm 2 に示す．

クラス毎に訓練データセットを処理していくことは，不均衡なデータセットにおける圧縮時のニューラルネットの学習を補助することに加え，実行時間の面でも有利である．MUNGE の実行速度のボトルネックになるのは，あるインスタンスについて各インスタンスとのユークリッド距離を導出し，最近傍を求める処理である．Adaptive MUNGE では，訓練データセットをクラス毎に分けて処理するため，ユークリッド距離を導出するために考慮するインスタンス数が減少し，実行速度が改善される．

図 1 は，単純な 2 次元分布（True Dist）と，True Dist から抽出された 450 点の訓練データから MUNGE，及び，Adaptive MUNGE によって生成された擬似データの分布を示している．また，波線は訓練データによって得られた

Algorithm 2 Adaptive MUNGE

Require:
訓練データセット T, イテレーション回数 k, 確率パラメータ p
Ensure: $k \times size(T)$ のクラスラベルなしデータセット D

$D \leftarrow \phi$
loop k 回:
  for all T のクラス T_c **do**
    $T'_c \leftarrow T_c$
    for all T'_c のインスタンス e **do**
      $e' \leftarrow T'_c$ 内の最近傍 e
      for all e の特徴量 a **do**
        p の確率で:
        if a が連続属性の場合 **then**
          $e_a \leftarrow e_a - u|e_a - e'_a| \quad u \sim U(0,1)$
          $e'_a \leftarrow e'_a - u|e_a - e'_a| \quad u \sim U(0,1)$
        else
          e の特徴量と e' の特徴量を入れ替える
        end if
      end for
    end for
    $D \leftarrow D \cup T'_c$
  end for
end loop

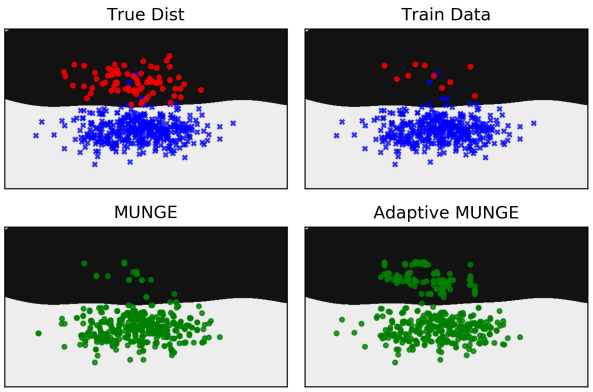

図 1 2 次元データにおける擬似データ生成手法の比較．波線は Train Data で訓練した SVM の決定境界を示す．Adaptive-MUNGE は，少数派クラスの擬似データを多く生成することで MUNGE に比べ真の分布を近似することができており，またクラスの不均衡も軽減されている．

SVM による決定境界を示している．

予想通り，MUNGE によって生成された擬似データは，多数派クラスのデータ付近の擬似データを多く生成していることが見て取れる．さらに擬似データの生成数を増やすと，少数派クラスの事前確率が下がり，データ全体の均衡度が下がる．Adaptive MUNGE は，クラス毎に擬似データを生成し，少数派クラス周辺の擬似データをより多く生成することで，MUNGE に比べ真の分布を近似することができている．

4. 実験と評価

4.1 データセット

本章では，ベンチマークに対して提案手法と既存手法の

表 1 データセット

データセット	特徴量	データ数	均衡度
LETTER	16	20000	0.0377
VEHICLE	18	846	0.2350
Mammography	6	11118	0.0232
COVTYPE	54	38501	0.0713

比較実験を行う．データセットの概要に関しては，表 1 にまとめた．表 1 における均衡度とは，訓練データにおけるクラスの均衡度合いを示す指標で，少数派クラスの事前確率である．本研究では，既存研究 [7] で用いられたデータセット，及び，不均衡学習における研究 [10], [11] で用いられているデータセットを用いて，実験を行なった．LETTER と VEHICLE，COVTYPE は UCI Repository[17] から取得した．Mammography は OpenML[18] から取得した．本研究では，既存研究 [7] に基づいて，バイナリ分類問題において比較実験を行うため，[7], [10], [11] に従って元のデータセットを修正した．LETTER は，クラス「O」を少数派クラス，それ以外の 25 文字を多数派クラスとすることでバイナリ問題に変換した．COVTYPE は，35754 サンプルと 2747 サンプルの 2 つのクラスを使用した．

4.2 実験手順と評価方法

アンサンブルの構築には様々なアルゴリズムを利用して多様なモデルを生成する．具体的には，SVM，ニューラルネットワーク，KNN，決定木，Bagged Decision Trees，Boosted Decision Trees，Boosted Decision Stumps を使用する．アルゴリズム毎に，様々なパラメータ設定を使用することで，総勢 844 のモデルを訓練し，アンサンブルの候補とする．いくつかのモデルは優れた性能を有すが，パフォーマンスが平均以下のモデルも存在する．本研究では，最適化に基づく枝刈りである RSE（式 (3)）と RSE-w（式 (4)）を使用しアンサンブルを構築し，各データセットにおいてパフォーマンスの良い方を採用する．RSE によって生成されたアンサンブルは，優れた般化性能を有する複雑なモデルであり，これをモデル圧縮の対象とする．MUNGE，Adaptive MUNGE は，それぞれの手法によって生成された擬似データで訓練された中間層が 128 個のユニットを持つ 2 層のニューラルネットを示す．ニューラルネットは，最適化アルゴリズムには Adam[19] を用い，バッチサイズは 128，活性化関数には ReLU[20]，出力層の活性化関数にはシグモイド関数を用いる．

各データセットに対して，5 分割交差検証によって実験を行う．また，5 分割交差検証における各訓練データを用いて，さらに 5 分割交差検証を行うことで，RSE のパラメータを決定する．評価は，アンサンブル，元の訓練データのみを用いて学習した最良の単一ニューラルネット，アンサンブルの候補に用いられているもののうち最良の単一モデル，MUNGE と，Adaptive MUNGE に対して行う．評価指標には，RMSE（Root Mean Squared Error），及び F 値を用いる．F 値は不均衡なデータに関する他の過去の研究で使用されており，クラスの不均衡に対して頑強であると考えられている [9].

表 2 擬似データ生成にかかった時間（秒）

データセット	MUNGE $mean \pm std$	Adaptive MUNGE $mean \pm std$	改善率（%）
LETTER	198.838790 ± 4.038973	91.071705 ± 3.192761	54.198
VEHICLE	9.102416 ± 0.092772	3.354137 ± 0.421760	63.151
Mammography	58.750733 ± 3.478732	29.389855 ± 0.504321	49.975
COVTYPE	552.654022 ± 5.301989	236.756285 ± 3.545175	57.160

4.3 実験結果と考察

図 2 は，各データセットに対するモデル圧縮の結果であり，5 分割交差検証に対する平均 F 値，平均 RMSE を示している．F 値は，高いほど不均衡なデータに対してうまく分類できていることを示し，RMSE は，その値が低いほどモデルのパフォーマンスが高いことを示している．それぞれに対して，最も良いパフォーマンスを示している水平線は訓練データに対するアンサンブルの結果である．また，best neural net は元の訓練データで訓練できる最良のニューラルネットを示し，best single model はアンサンブルの候補となる最良の単一モデルの平均パフォーマンスを示す．Adaptive MUNGE は，F 値と RMSE 両方において，MUNGE より優位にあり，均衡度が低いデータセットに関しては差が顕著である．Adaptive MUNGE によって少数派クラスの割合が増えるように擬似データを生成することで，不均衡データに対しても圧縮後のモデルが過学習することなく，アンサンブルの決定境界を近似できたと考えられる．擬似データセットが 800k を越えるとアンサンブルの最良の候補モデルよりも良い精度で分類できている．また，Adaptive MUNGE で生成した 100k の擬似データで訓練したニューラルネットは，元のアンサンブルとほぼ同等の RMSE，及び，F 値であることがわかる．しかし，COVTYPE は均衡度が低いが，あまり差がない．これは COVTYPE には離散属性が多く，MUNGE と Adaptive MUNGE の処理に差がつかないためだと考えられる．

表 2 は，各データセットにおいて，24 コアの CPU を使用し，訓練データから 100k の擬似データ生成にかかった時間を示している．Adaptive MUNGE は，MUNGE で擬似データ生成にかかる時間を平均して半減できている．Adaptive MUNGE は，入力データに対してクラス毎に処理することで，MUNGE においてボトルネックであった近傍を求める処理において，距離計算の対象になるデータ数を分割することで，擬似データ生成にかかる時間を半減できたと考えられる．

5. 結論

本研究では，いかなるアンサンブルにもモデル圧縮が可

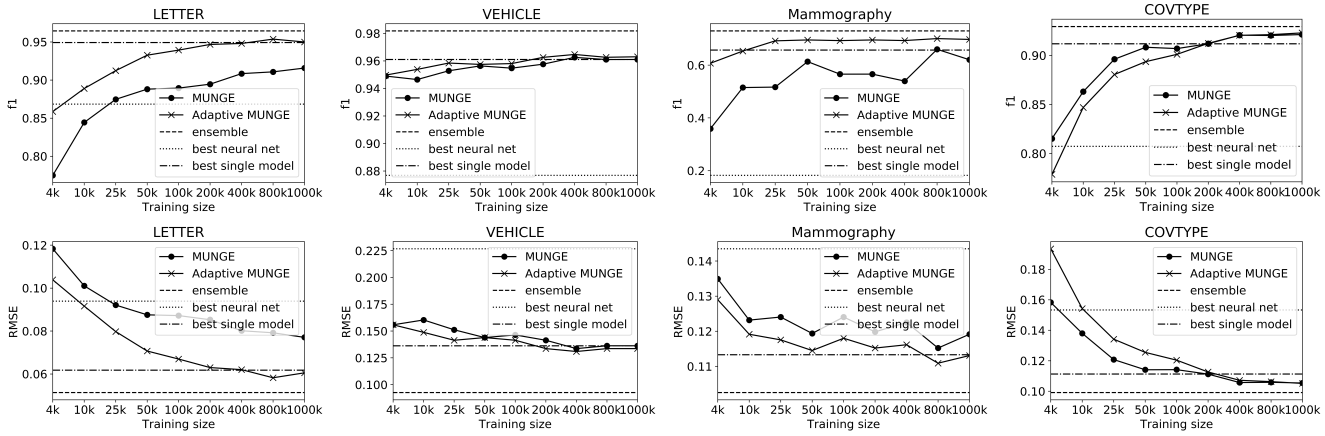

図 2 各データセットにおけるモデル圧縮の結果．ensemble は訓練データに対するアンサンブルの結果であり，best neural net は元の訓練データで訓練できる最良のニューラルネット，best single model はアンサンブルの候補となる最良の単一モデルの平均パフォーマンスを示す．Adaptive MUNGE は MUNGE に比べ，均衡度の低いデータ（LETTER, Mammography）に対して，良いパフォーマンスを発揮している．

能な擬似データ生成手法を提案した．提案手法 Adaptive MUNGE では，入力データをクラスごとに処理することで，元のデータの不均衡を改善した擬似データを生成する．既存手法 MUNGE との比較実験により，訓練データの均衡度が低いほど，圧縮後モデルの過学習を防ぎ，良いパフォーマンスを達成できることを明らかにした．また，Adaptive MUNGE は，不均衡データに対してモデル圧縮のパフォーマンスを向上させるだけでなく，擬似データ生成を高速化し，さらにユーザが実験によって定めなければならないパラメータが少ない．比較実験により，Adaptive MUNGE は，MUNGE に対して擬似データ生成にかかる時間を半減できることを示した．本実験ではバイナリ分類問題のみを扱ったが，Adaptive MUNGE はクラス毎の不均衡を改善する効果があるため，多クラス分類問題においてさらにその効果を発揮すると考えられる．他クラス分類や回帰問題への適用は今後の展望である．

謝辞 本研究の一部は，JSPS 科研費（課題番号 16H02904）の助成によって行われた．

参考文献

[1] Dietterich, T. G.: Ensemble Methods in Machine Learning, *Proc. MCS*, pp. 1–15 (2000).
[2] Bauer, E. and Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, *Machine Learning*, Vol. 36, No. 1, pp. 105–139 (1999).
[3] Opitz, D. and Maclin, R.: Popular Ensemble Methods: An Empirical Study, *JAIR*, Vol. 11, pp. 169–198 (1999).
[4] Zhou, Z.-H.: *Ensemble Methods: Foundations and Algorithms*, Chapman & Hall/CRC, 1st edition (2012).
[5] Dietterich, T. G.: Machine-Learning Research: Four Current Directions, *AI Magazine*, Vol. 18, No. 4, pp. 97–136 (1997).
[6] Zeng, X. and Martinez, T. R.: Using a Neural Network to Approximate an Ensemble of Classifiers, *Neural Processing Letters*, Vol. 12, No. 3, pp. 225–237 (2000).
[7] Buciluǎ, C., Caruana, R. and Niculescu-Mizil, A.: Model Compression, *Proc. ACM SIGKDD*, pp. 535–541 (2006).
[8] Tony, L.: Model Based Sampling - Fitting an Ensemble of Models into a Single Model, *Proc. CSCI*, pp. 186–191 (2015).
[9] Liu, A., Ghosh, J. and Martin, C. E.: Generative Oversampling for Mining Imbalanced Datasets., *DMIN*, pp. 66–72 (2007).
[10] Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P.: SMOTE: Synthetic Minority Oversampling Technique, *J. Artif. Int. Res.*, Vol. 16, No. 1, pp. 321–357 (2002).
[11] He, H., Bai, Y., Garcia, E. A. and Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning., *IJCNN*, pp. 1322–1328 (2008).
[12] Caruana, R., Niculescu-Mizil, A., Crew, G. and Ksikes, A.: Ensemble Selection from Libraries of Models, *Proc. ICML* (2004).
[13] Zhou, Z.-H., Wu, J. and Tang, W.: Ensembling neural networks: Many could be better than all, *Artif. Intell.*, Vol. 137, No. 1, pp. 239 – 263 (2002).
[14] Li, N. and Zhou, Z.-H.: Selective Ensemble under Regularization Framework, *Proc. MCS*, pp. 293–303 (2009).
[15] Tsoumakas, G., Partalas, I. and Vlahavas, I.: An Ensemble Pruning Primer, *SUEMA*, pp. 1–13 (2009).
[16] Hernández-Lobato, D., Martínez-Muñoz, G. and Suárez, A.: Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles, *Neurocomputing*, Vol. 74, No. 12, pp. 2250 – 2264 (2011).
[17] Lichman, M.: UCI Machine Learning Repository (2013).
[18] Vanschoren, J., van Rijn, J. N., Bischl, B. and Torgo, L.: OpenML: Networked Science in Machine Learning, *SIGKDD Explorations*, Vol. 15, No. 2, pp. 49–60 (2013).
[19] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization., *CoRR*, Vol. abs/1412.6980 (2014).
[20] Glorot, X., Bordes, A. and Bengio, Y.: Deep Sparse Rectifier Neural Networks, *Proc. AISTATS*, Vol. 15, pp. 315–323 (2011).