情報処理学会論文誌　データベース　Vol.11　No.2　7–15 (July 2018)

モデル圧縮におけるクラス不均衡に着目した疑似データ生成手法の提案

河野 晋策[1,a)]　若林 啓[2,b)]

受付日 2017年12月5日, 採録日 2018年4月6日

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疑似データを生成して近似モデルの学習に用いるが，この疑似データ生成において真のデータ分布をよく近似した疑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疑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疑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．

キーワード：モデル圧縮，データ生成，疑似データ，機械学習，アンサンブル，ニューラルネット，分類，不均衡学習

Proposal of Synthetic Data Generation Method Focusing on Class Imbalance in Model Compression

Shinsaku Kono[1,a)]　Kei Wakabayashi[2,b)]

Received: December 5, 2017, Accepted: April 6, 2018

Abstract: In classification by machine learning, an ensemble method that is a combination of multiple classification models is often highly accurate. However, since an ensemble method require a large amount of computational resources, it is difficult to use it in an environment with limited computing resources such as mobile phones. To solve this problem, model compression that approximates an ensemble with a small neural network has been proposed. In model compression, a large amount of synthetic data is generated based on the original data and used for learning of the approximate model. It is important to obtain a synthetic data distribution that closely approximates the true data distribution in this synthetic data generation in order to bring the performance of the approximate model closer to the original ensemble. In this study, we propose Adaptive MUNGE which generates synthetic data which is effective for learning of approximate model faster than existing method by considering the distribution bias of each classification class. Experimental results show that the proposed method can generate synthetic data at high speed compared with the existing method and can realize model compression that maintains more accuracy.

Keywords: model compression, data generation, synthetic data, machine learning, ensemble method, neural network, classification, imbalanced learn

[1] 筑波大学情報学群知識情報・図書館学類
College of Knowledge and Library Sciences, School of Informatics, University of Tsukuba, Tsukuba, Ibaraki 305–8550, Japan
[2] 筑波大学図書館情報メディア系
Faculty of Library, Information and Media Science, University of Tsukuba, Tsukuba, Ibaraki 305–8550, Japan
[a)] s1411524@klis.tsukuba.ac.jp
[b)] kwakaba@slis.tsukuba.ac.jp

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．たとえば，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，以下の訓練フェーズ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) 訓練フェーズ
  (a) 訓練データに対して，アンサンブルを学習
  (b) 訓練データを元に疑似データを生成
  (c) 1a で訓練したアンサンブルを用いて 1b で生成した疑似データにクラスラベルを付与
  (d) 1c のクラスラベル付き疑似データを用いて，単一モデルを訓練
(2) 実行フェーズ
  (a) 1d で得られた単一モデルを本番環境にデプロイ
  (b) 2a でデプロイされたモデルを用いて予測

訓練フェーズは訓練用サーバを用いて行い，実行フェーズは，軽量デバイス上で行う．実行フェーズにおける本番環境は，携帯端末など計算資源の限られた環境を指す．アンサンブルによってクラスラベルを付与された疑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疑似データを生成する 1b 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．

疑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，その距離に応じた分散の正規分布を用いて疑似データを生成する．しかし，正規分布を用いるため，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疑似データを生成することで，データの分布により忠実な疑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疑似データ生成において，アンサンブル中のアルゴリズムの構造に依存しない 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，分類クラスごとに疑似データを生成することで，疑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なくすことで，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．さらに，分類クラスごとに疑似データを生成することで既存手法のボトルネックを改善し，提案手法はより高速に疑似データを生成することを示す．

本稿の構成は以下のとおりである．2 章で本研究と関連するモデル圧縮，および，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 章で提案手法の Adaptive MUNGE のアルゴリズムを示す．4 章で提案手法と既存手法の比較実験を行ったうえで，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，さかんに行われている [6], [7], [8]．Zeng ら [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疑似データとして用いている．しかし，Zeng らは，疑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付けた．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，より多くの疑似データが必要であることを示している．Buciluǎ らは，Zeng ら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うえで，複雑なアンサンブルのモデル圧縮を行うためには，より多くの疑似データが必要であることを示した．この疑似データを生成するために，Buciluǎ らは，疑似データ生成手法である MUNGE を提案した．

Algorithm 1 MUNGE

Require:
訓練データセット（クラスラベルなし）T，疑似データ生成数 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
Output: クラスラベルなしデータセット D

```
D ← φ
while |D| < k do
    T' ← T
    for all T' のインスタンス e do
        e' ← T' 内の最近傍 e
        for all e の特徴量 a do
            t ∼ U(0, 1)
            if t ≤ p then
                if a が連続属性の場合 then
                    sd ← |e_a − e'_a|/s
                    e_a ← Norm(e'_a, sd), e'_a ← 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 if
        end for
    end for
    D ← D ∪ T'
end while
```

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．MUNGE は，アンサンブル中のアルゴリズムの構造に依存しないモデル圧縮における疑似データ生成に利用できるが，以下のような問題点がある．

- 訓練データのクラス分布に偏りが存在する場合，疑似データ生成により，さらにクラスの偏りが顕著になる．
- クラスの境界線を考慮しないため，クラスの境界に当たる疑似データを生成し，モデルの学習の妨げになることがある．

Lindgren [8] は，モデル圧縮における疑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，MUNGE における問題点を解決し，アンサンブル中のアルゴリズムの構造に依存しないモデル圧縮が適用可能な疑似データ生成手法を提案する．

モデル圧縮における疑似データ生成は，不均衡データ学習におけるオーバ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，多くの疑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，すで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つながらないモデルは，排除される．このように候補モデルのすべ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，および，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，計算時間に鑑み，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．最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 ら [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．このとき，$w_i \geq 0$ 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．

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

このとき，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，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. Adaptive MUNGE による疑似データ生成

本研究では，MUNGE よりも訓練データの分布をよりうまく近似する疑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，疑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あ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．与えられるデータセットにおいて，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ものと仮定する．データセットのクラスラベル T_c の各インスタンス 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．特徴量が連続属性の場合，式 (5) を用いて，疑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 の間に来るように疑似データの値を得ることができる．また，乱数 $u\sim U(0,1)$ を用いることで疑似データをインスタンス e とその最近傍 e'_a の間のランダムな位置に配置し，疑似データの多様性を担保することができる．さらに，u の導入により 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 e_a と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．

前述のように，MUNGE は各訓練データに対して最近傍を発見し，その距離に応じた分散の正規分布を用いて疑似データを生成する．しかし，正規分布を用いることによって，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．Adaptive MUNGE では，式 (5) のように各訓練データと最近傍の間に疑似データを生成することで，この問題を解決する．より直感的な理解のため，簡単なデータセットに対する MUNGE と Adaptive MUNGE の差を**図 1** に示す．

また，Adaptive MUNGE では，疑似データ生成数を決めるパラメータ k に対して，クラスごとのイテレーション回数を定め，クラスごとの疑似データ生成数を決める．これによりクラス間の不均衡を改善することができる．不均衡データに対し，少数派クラスをオーバ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]．これら研究に基づいて，本研究では少数派クラスの周辺のデータを多く生成することで不均衡データに対して，先行研究よりも圧縮後モデルのパフォーマンスを向上することを期待する．このとき，疑似データ数の自由度は，境界面の構造に影響を与えると考えられる．たとえば，クラスごとのデータ数の比を均等にした場合，元の比を保つ場合よりも少数派クラスの決定領域が拡大する可能性がある．しかし，ニューラルネットや SVM といった識別モデルにおいては，クラスごとのデータ数が誤差関数に与える影響は，決定境界と各データとの距離が変化することによる影響に比べて小さい．このため，モデ

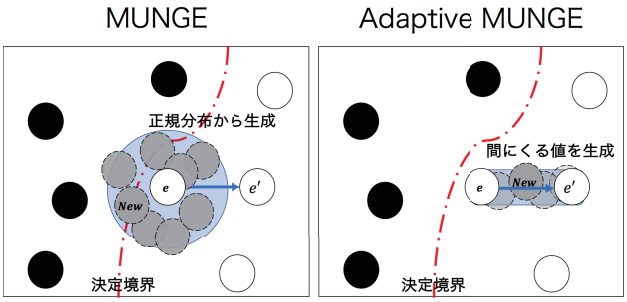

図 1　3 次元データにおける MUNGE および Adaptive MUNGE の疑似データ生成の違い．各データは特徴量として，2 つの連続属性と「黒」か「白」のクラスラベルを持つ．MUNGE は正規分布を用いるため，分類タスクにおけるクラスの境界に疑似データを生成してしまう可能性がある．Adaptive MUNGE は e と e' の間に疑似データを生成することでその問題点を解決する

Fig. 1　Difference in pseudo data generation of MUNGE and Adaptive MUNGE in 3D data. Each data has 2 continuous attributes and a class label of "black" or "white" as a feature. Since MUNGE uses normal distribution, there is a possibility that pseudo data is generated at the class boundary in the classification task. Adaptive MUNGE solves this problem by generating pseudo data between e and e'.

Algorithm 2 Adaptive MUNGE

Require:
　訓練データセット T，疑似データ生成数 k，確率パラメータ p
Output: クラスラベルなしデータセット D

$D \leftarrow \phi$
$sampleSize \leftarrow k/$(T のクラス数)
for all T のクラス T_c **do**
　$D_c \leftarrow \phi$
　while $|D_c| < sampleSize$ **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**
　　$t \sim U(0,1)$
　　if $t \leq p$ **then**
　　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 if
　　end for
　　end for
　　$D_c \leftarrow D_c \cup T'_c$
　end while
　$D \leftarrow D \cup D_c$
end for

ル圧縮の単一モデルにニューラルネットを用いる提案手法においては，疑似データ数の自由度による境界面への影響は小さいと考えられる．Adaptive MUNGE のアルゴリズムを Algorithm 2 に示す．

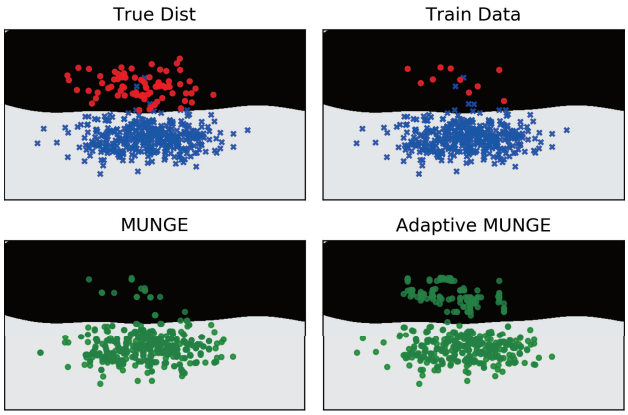

図 2　2 次元データにおける疑似データ生成手法の比較．波線は Train Data で訓練した SVM の決定境界を示す．Adaptive MUNGE は，少数派クラスの疑似データを多く生成することで MUNGE に比べ，クラスの不均衡が軽減されている

Fig. 2　Comparison of pseudo data generation method in 2D data. The wavy line shows the decision boundary of the SVM trained with Train Data. Adaptive MUNGE reduces class imbalance compared with MUNGE by generating many pseudo data of minority class.

クラスごとに訓練データセットを処理することは，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 は，訓練データセットをクラスごとに分割し，疑似データ生成することで，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．また，Adaptive MUNGE では，疑似データ生成数を決めるパラメータ k に対して，クラスごとの疑似データ生成数の均衡がとれるようにクラスごとのイテレーション回数を定める．このため，少数派クラスにはイテレーション数が多く割り当てられ，一方，多数派クラスには少なく割り当てられる．不均衡データでは，少数派クラスに含まれるインスタンス数の訓練データ全体に占める割合は，非常に小さい．このとき，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も少ないので，イテレーション数が多くても全体として実行速度が改善される．

図 2 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，および，Adaptive MUNGE によって生成された疑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 SVM による決定境界を示している．

MUNGE によって生成された疑似データは，多数派クラスのデータ付近の疑似データを多く生成していることが見てとれる．Adaptive MUNGE は，クラスごとに疑似データを生成し，少数派クラス周辺の疑似データをより多く生

成することで，MUNGE に比べクラス間の不均衡を改善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，および，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，文献 [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，35,754 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．アルゴリズムごとに，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，2.2 節で述べた 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．

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．別の 5 分割交差検証を行い，MUNGE と Adaptive MUNGE のパラメータを定める．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．MUNGE, Adaptive MUNGE は，それぞれの手法によって生成された疑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．評価指標には，RMSE（Root Mean Squared Error），および 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]．

実験では，CPU に Intel Xeon E5-2420 [*1]（6 コア）2 機，メモリに DDR3-1333 96GB を搭載したサーバを訓練用サーバとして用いた．

表 1　データセット
Table 1　Description of datasets.

データセット	特徴量	データ数	均衡度
LETTER	16	20,000	0.0377
VEHICLE	18	846	0.2350
Mammography	6	11,118	0.0232
COVTYPE	54	38,501	0.0713

4.3 実験結果と考察

図 3 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，LETTER, VEHICLE，Mammography の 3 つのデータセットに対して，F 値と RMSE 両方において MUNGE より優位にある．また，均衡度が低いデータセットに関してはその差が顕著である．COVTYPE に関しても，疑似データ数が少ないときは MUNGE が優っているが，ベストスコアに関しては，Adaptive MUNGE が優っている．Adaptive MUNGE によって少数派クラスの割合が増えるように疑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．疑似データのサイズが 800k を超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 の疑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，および，F 値であることが分かる．しかし，COVTYPE は均衡度が低いが，MUNGE と Adaptive MUNGE の性能にあまり差がない．これは

[*1] https://ark.intel.com/products/64617/Intel-Xeon-Processor-E5-2420-15M-Cache-1_90-GHz-7_20-GTs-Intel-QPI

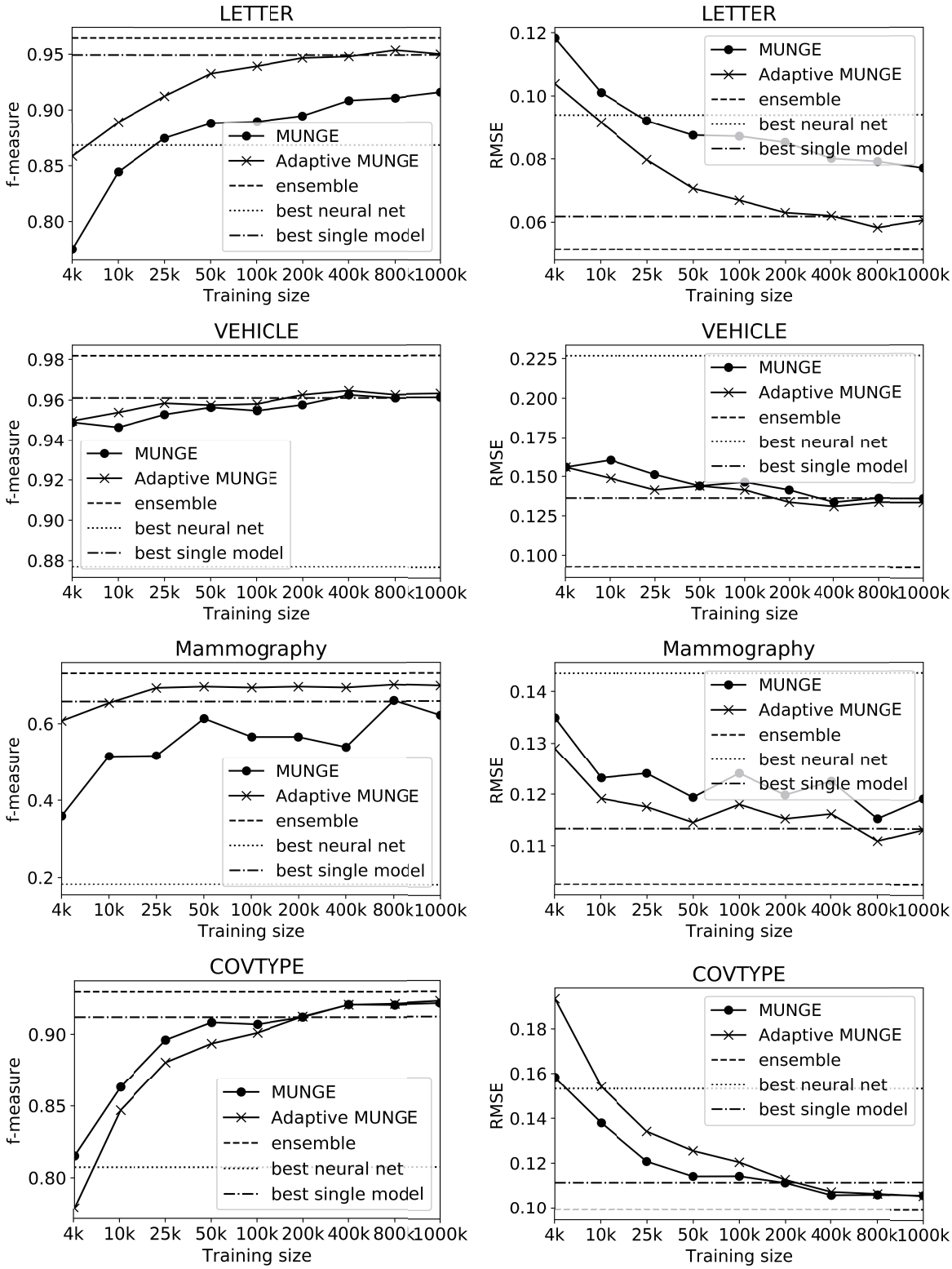

図 3　各データセットにおけるモデル圧縮の結果．ensemble は訓練データに対するアンサンブルの結果であり，best neural net は元の訓練データで訓練できる最良のニューラルネット，best single model はアンサンブルの候補となる最良の単一モデルの平均パフォーマンスを示す．Adaptive MUNGE は MUNGE に比べ，均衡度の低いデータ（LETTER，Mammography）に対して，良いパフォーマンスを発揮している

Fig. 3　The result of model compression in each data set. “ensemble” indicates the average performance of the ensemble for the training data. “best neural net” shows the average performance of the best neural net that can be trained with the original training data. “best single model” shows the average performance of the best single model that is a candidate for the ensemble. Compared to MUNGE, Adaptive MUNGE shows good performance against data with imbalanced data (LETTER, Mammography).

表 2　各データセットにおけるモデル圧縮の結果（f-measure）
Table 2　f-measure results of model compression in each data set.

データセット	ensemble	best neural net	best single model	MUNGE	Adaptive MUNGE
LETTER	0.96451	0.86841	0.94923	0.91577	0.95373
VEHICLE	0.98180	0.87680	0.96099	0.96256	0.96468
Mammography	0.73004	0.18201	0.65743	0.66022	0.70087
COVTYPE	0.92937	0.80736	0.91185	0.92124	0.92290

表 3　各データセットにおけるモデル圧縮の結果（RMSE）
Table 3　RMSE results of model compression in each data set.

データセット	ensemble	best neural net	best single model	MUNGE	Adaptive MUNGE
LETTER	0.05135	0.09391	0.06179	0.07707	0.05826
VEHICLE	0.09258	0.22678	0.13627	0.13363	0.13093
Mammography	0.10253	0.14349	0.11332	0.11522	0.11091
COVTYPE	0.09911	0.15335	0.11130	0.10542	0.10512

表 4　疑似データにおける少数派クラスの割合
Table 4　Ratio of minority class in pseudo data.

データセット	MUNGE	Adaptive MUNGE
	mean ± *std*	*mean* ± *std*
LETTER	0.0378 ± 0.00017	0.4941 ± 0.00036
VEHICLE	0.2322 ± 0.00535	0.4793 ± 0.01829
Mammography	0.0203 ± 0.00028	0.4387 ± 0.00277
COVTYPE	0.0694 ± 0.00015	0.4840 ± 0.00136

表 5　疑似データ生成にかかった時間（秒）
Table 5　Time in seconds to generate pseudo data.

データセット	MUNGE	Adaptive MUNGE	改善率
	mean ± *std*	*mean* ± *std*	(%)
LETTER	198.839 ± 4.039	91.072 ± 3.193	54.198
VEHICLE	9.102 ± 0.093	3.354 ± 0.422	63.151
Mammography	58.751 ± 3.479	29.390 ± 0.504	49.975
COVTYPE	552.654 ± 5.302	236.756 ± 3.545	57.160

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，**表 3** は 4 つのベンチマークそれぞれに対して，圧縮対象のアンサンブル，元の訓練データで訓練できる最良のニューラルネット，アンサンブルの候補となる最良の単一モデル，および，MUNGE，Adaptive MUNGE のベストスコアを示す．各値は，5 分割交差検証における平均パフォーマンスである．すべてのベンチマークにおいて Adaptive MUNGE は，つねに MUNGE の性能を上回り，圧縮においての有効性を示している．また，均衡度の低いデータセットに関しては，その差が顕著に表れている．

表 4 は，各データセットにおいて，訓練データから生成した 100k の疑似データにおける少数派クラスの割合を示している．具体的には，訓練データを用いて訓練したアンサンブルが，訓練データの少数派クラスに属すると予測した疑似データの割合である．MUNGE で生成した疑似データは元の訓練データの事前確率に非常に近い値を示している．それに対して，Adaptive MUNGE は狙いどおり少数派クラスのデータを多く生成することで均衡のとれた疑似データを生成することができている．

表 5 は，各データセットにおいて，訓練データから 100k の疑似データ生成にかかった時間を示している．Adaptive MUNGE は，MUNGE で疑似データ生成にかかる時間を平均して半減できている．前述のとおり Adaptive MUNGE は，入力データに対してクラスごとに処理する．これにより，距離計算の対象になるデータ数を分割することで，問題のスケールを小さくすることができる．その結果，MUNGE においてボトルネックを改善し，疑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，アンサンブル中のアルゴリズムの構造に依存しないモデル圧縮が可能な疑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疑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，疑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 に対して疑似データ生成にかかる時間を半減できることを示した．Adaptive MUNGE はクラスラベルに基づいて，訓練データを分割し疑似データを生成する．これにより，MUNGE においてボトルネックであった近傍を求める処理の問題のスケールを小さくすることができる．この「分割」の処理は，訓練データのクラス数に基づいて行われるので，多クラス分類問題では分割数が増加し，さらに問題のスケール

を小さくすることができる．よって，Adaptive MUNGEは，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．多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] Lindgren, T.: 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).

河野 晋策

2018 年筑波大学卒業．学士（図書館情報学）．同年株式会社リクルート入社．株式会社リクルートテクノロジーズ出向．

若林 啓　（正会員）

2012 年法政大学大学院博士課程修了．博士（工学）．同年筑波大学図書館情報メディア系助教．機械学習の研究に従事．日本データベース学会，ACM 各正会員．

（担当編集委員　荒巻 英治）