データセット多様性のソフトウェア・テスティング

中島 震

統計的な機械学習には通常のソフトウェア・テスティングが前提とする確定的なオラクルが存在しない．相対的な正解値に基づく疑似オラクルの方法であるメタモルフィック・テスティングを適用する．機械学習プログラムへの入力となるデータセットは多数のデータの集まりである．データ分布の違いが入力の多様性を表すといってよい．本稿では，入力データセットを系統的に生成する新しいメタモルフィック・テスティングの方法を提案する．具体例として，サポートベクタマシンならびにニューラルネットへの適用事例を報告する．

Deterministic oracle, used in standard software testing, is not available for statistical machine learning computer programs. Alternatively, metamorphic testing, pseudo oracle in view of data diversity, is adopted. Statistical machine learning, however, works on a dataset consisting of many data. Their distributions in the dataset show the diversity in testing. This paper proposes a new metamorphic testing method to take into account the dataset diversity. The method was applied to two well-known machine learning of support vector machines and neural networks.

1 はじめに

ビッグデータ分析や機械学習は，多数のデータからなるデータセットを対象とし，統計的な方法を用いてデータ間に内在する関係を推定する．自明でない有用な関係を見出すこと，新たなデータに対する結果を予測することが目的．得たい結果（正解値）が未知なことから，テスト不可能プログラム [7] といえる．

正解値が既知でないプログラムを検査する方法として，他プログラムの結果を正解値の代わりに用いる疑似オラクルが知られている．このような疑似オラクルへのアプローチとして，デザイン多様性とデータ多様性の 2 つがある．データ多様性[1] は系統的な入力データ生成に着目する考え方で，メタモルフィック・テスティング [5] などの方法がある．

データ多様性は，数値計算プログラムを検査する入力生成の方法として考案された．統計的な機械学習も数値計算を中心とする最適化問題として定式化される．実際，メタモルフック・テスティングは分類学習プログラムの検査に適用された（たとえば[8]）．ところが，機械学習プログラムへの入力は単独のデータではなく，データセットである．データ分布の違いが入力の多様性を表すといってよい．異なるデータ分布を系統的に生成することで，さまざまな入力データセットを得ることができる [6]．つまり，データセット多様性に着目すべきである．

本稿では，文献[6] の方法を一般化し，データセット多様性を考慮したフォローアップ入力生成の新しい方法を提案する．サポートベクタマシンならびにニューラル・ネットという統計的な機械学習の代表的な方法への適用事例を報告する．

2 ソフトウェア・テスティング

2.1 確定的なオラクル

ソフトウェア・テスティングの関心事は，検査対象プログラム中の欠陥を如何にして効率よく発見するかである．今，検査対象を関数 $f(x)$ と表す．入力データ X に対する既知の正解値 C^X を用いて，$f(X) = C^X$ が成り立つかを調べる．正解値は設計仕様や理論値か

---

* Software Testing under Dataset Diversity.

Shin Nakajima, 国立情報学研究所, National Institute of Informatics.

ら決まる．検査対象プログラムからみると，絶対的な正しさを与える．

計算結果 $f(X)$ が C^X に一致しない時，プログラムに不具合がある．欠陥箇所を実行しない可能性が残るので，一致する場合でも不具合がないとはいえない．入力を工夫することで，不具合の存在がわかると期待する．ソフトウェア・テスティングの主要な課題は，プログラムの可能な実行経路をできるだけカバーすること（カバレッジ），未知の欠陥がありそうな経路を実行する入力を得ること（コーナーケース検査）の 2 つになる．

一般に，テスト・オラクルは，計算結果と比較可能な正解値の求め方ならびに正解値と検査する方法を指す．検査対象を決めた時，テスト・オラクルは入力に対して受理あるいは棄却を判断する．確定的なオラクルは，同一入力に対して常に同じ判断結果を返す†1．本稿で考察する統計的な機械学習ソフトウェアには確定的なオラクルが存在しない．このようなソフトウェアに対する検査の信用レベルを如何にして高めるかが課題である．

2.2 疑似オラクル

入力 X に対する正解値 C^X が事前に知られていない場合を考える．一般に，欲しい値が未知のプログラムは正解値がわからない．知っていればプログラムを作る必要がない．このようなプログラムの検査では，相対的な正解値を C^X の代わりに使う疑似オラクルを用いる [7]．

相対的な正解値は，検査対象以外の他プログラムが計算した値（黄金出力）である．プログラムは正しさが保証されていないので，その計算値は絶対的に正しいとはいえない．プログラムを改版した際に実施する回帰テストで，改版前プログラムと同じ結果が得られれば，改版後プログラムに新たな欠陥が混入していない，と考えることに似ている．

相対的な正解値を得る方法として，デザイン多様性とデータ多様性の 2 つの考え方がある．デザイン多様性に基づく方法は，同じ機能仕様を満たすプログラムを複数開発するもので，N バージョン・プログラミングがある．独立した開発チームが異なるプログラミング言語を用いる等の工夫によって，疑似オラクルの信用レベルを向上させる．

データ多様性 [1] は，数値計算など多様なデータによる検査が必要なプログラムで有用な考え方で，メタモルフィック・テスティング [5] がある．これは，検査対象プログラム $f(X)$ だけを用いる疑似オラクルといえる．入力 $X^{(m)}$ に対する計算結果を $f(X^{(m)})$ とする．$X^{(m)}$ に変換 T を施してフォローアップ入力 $X^{(m+1)} = T(X^{(m)})$ を得る．ここで，変換 T は 2 つの計算結果の間に適切なメタモルフィック関係 Rel^T が成り立つように，検査対象プログラムの機能仕様から見出す．2 つの計算結果が，$Rel^T(f(X^{(m)}),\ f(X^{(m+1)}))$ を満たさない時，f に不具合があると結論する．

3 機械学習ソフトウェアのテスティング

3.1 教師あり機械学習

教師あり機械学習の基本的な考え方を，最適化の問題として説明する．今，データセット D をデータ点 $\langle \vec{x}^n, y^n \rangle$ の集まりとする．ここで，$\vec{x}^n$ は多次元ベクタ，y^n は教師ラベルを表す．

統計的な機械学習は，パラメータ θ で特徴つけられた関数 $F^C(\theta;\ \vec{x})$ の集まり $\{F^C(\theta;\ \vec{x})\}^\theta$ から，与えられたデータセット D に対する目的関数 $\mathcal{E}(\theta;\ \{\langle \vec{x}^n, y^n \rangle\})$ を最小にするパラメータ θ^* を求めること†2．つまり，$\theta^* = arg\,min\ \ \mathcal{E}(\theta;\ \{\langle \vec{x}^n, y^n \rangle\})$ である．関数 $F^C(\theta^*;\ \vec{x})$ を決めるには，ハイパーパラメータ C の値を決めておく必要があり，θ^* は C に依存する．

パラメータ推定に用いるデータセットを訓練データセット D_R，得られた関数 $F^C(\theta^*;\ \vec{x})$ の評価に用いるものを試験データセット D_T，最適なハイパーパラメータの推定に用いるものを検証データセット D_V と呼ぶ．何らかの方法で決めたハイパーパラメータ値を C_o とする時，推定したパラメータ $\theta^*(C_o)$ の良し悪しは，検証データセット D_V に関数 F を適用した結果として得られる正解率で評価する．

†1 本稿では検査対象が停止しない場合は考えない．

†2 制約条件 $\mathcal{C}(\theta;\ \{\langle \vec{x}^n, y^n \rangle\})$ を伴うこともある．

3.2 メタモルフィック・テスティングの適用

機械学習のソフトウェア・テスティング対象は，θ^* を求める最適化問題の解法プログラム f である．ハイパーパラメータ C は定数であり，検査対象プログラム f は訓練データセット D_R を入力として，パラメータ θ^* を計算する．θ^* は未知なので，メタモルフィック・テスティングを用いる．

機械学習タスクを定式化した最適化問題は目的関数 $\mathcal{E}$ ならびに制約条件 $\mathcal{C}$ の形で数式で表される．ソフトウェア工学の観点では，最適化問題が宣言的な要求仕様に対応するとしてよい．今，目的関数 $\mathcal{E}$ を不変にするようなフォローアップ変換 T を選べば，メタモルフィック関係が自明になる．あるいは，見通しの良い Rel^T が成り立つように T を選ぶ．メタモルフィック・テスティングを分類学習に応用した事例研究 [8] は，表 1 に示す一般的なメタモルフィック性を整理した．直感的に明らかであるが，変換関数の具体的な導出方法は示されていない．文献 [6] は，サポートベクターマシンについて，最適化問題の形式からフォローアップ変換を導出する系統的な方法を論じた．

3.3 データセット多様性

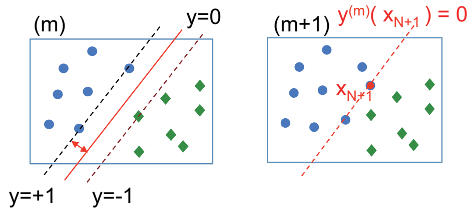
検査対象 f は非線形の目的関数 $\mathcal{E}$ を解く数値計算プログラムで，与えられた入力データセット D_R の要素全体に対する処理として実現される．単純な制御構造を持つ繰り返し処理であることから，制御フローグラフを対象とする通常のテスト・カバレッジは自ずから満たされ，良い基準にならない [6]．

一方，f の検査結果は入力データセット D_R に依存する．より詳しくは，学習タスク毎に，D_R 中のデータ点分布が計算結果に大きな影響を与える．一般に，データセットは無限にあるので，ランダム生成した分布のデータセットを用いても検査効率が悪い．対象学習タスクの性質に依存して適切な偏りを持つ分布を用いたい．

文献 [6] では，教師あり分類学習タスクを対象とするデータセット・カバレッジを導入した．関数 f を検査するという観点から，入力とする訓練データセット D_R 内のデータ分布を工夫するということ．分類を困難にするような限界データ点を系統的に導入する方法を示した．

表 1　一般的なメタモルフィック性

Additive	データ点の属性に加算
Multiplicative	データ点の属性に乗算
Permutative	データ点の入れ替え
Invertive	正解ラベルの反転
Inclusive	新しいデータ点の追加
Exclusive	既存データ点の削除



(a) m-th hyperplane　　(b) inclusive

図 1　Support Vector Machine

本稿では，文献 [6] の方法を一般化して，データセット内のデータ分布に注目する見方をデータセット多様性と呼び，データセット多様性を考慮したメタモルフィック・テスティングの一般的な枠組みを提案する．

テスト入力として用いる訓練データセットを $D^{(m)}$ と表記する．$D^{(0)}$ を何らかの方法で得た最初の訓練データセットとする．従来の方法では，フォローアップ入力は $D^{(m+1)} = T(D^{(m)})$ で求めた．一方，本稿では，フォローアップ入力を得る変換関数 T を拡張し，$T(D^{(m)},\ f(D^{(m)}))$ とする．フォローアップ入力 $D^{(m+1)}$ を得るのに，計算結果 $f(D^{(m)})$ を用いる．

検査は，関係 $Rel^T(f(D^{(m)}),\ f(D^{(m+1)}))$ を調べる．一般に，パラメータ数は膨大であり，全ての値を対象とする検査は容易でない．関数 g を $f(D)$ の計算結果 θ^* から検査に必要な値を抽出する処理とし，$Rel^T(f\circ g(D^{(m)}),\ f\circ g(D^{(m+1)}))$ とする．

4　適用事例

4.1　サポートベクターマシン

4.1.1　ラグランジュ形式

サポートベクタマシン（SVM）は教師あり分類学習の方法．データ点は多次元ベクタとスカラ値をとる

ラベルの組で，分類問題は N 個のデータ点からなるデータセットが与えられた時，ラベル値の異なるデータ点を分離するような境界を求めること．SVM は，求める分離境界の最も近くにあるデータ点（サポートベクタ）を発見し，サポートベクタとの距離（マージン）が最大となるように分離境界を決定する．図 1 ではラベルを図形で区別し，実線で分離境界線を，破線でサポートベクタ上の線を表した †3.

SVM は制約条件あり最適化問題として定式化されている．$\langle \vec{x}^n, \ell^n \rangle$ をデータ点，$\vec{x}^n$ を K 次元ベクタ，ℓ^n を -1 か $+1$ の値をとるラベルとする．今，$y(\vec{x}) = \vec{w}^T \cdot \vec{x} + b$ とおくと，分離超平面は $y(\vec{x}) = 0$ を満たす．2 つのパラメータ $\vec{w}$ と b を求める．

$$arg\,\min_{\vec{w},b}\ \frac{1}{2}\|\ \vec{w}\ \|^2 \qquad \textbf{s.t.}\ \ \ell^n(\vec{w}^T\cdot\vec{x}^n + b) \geq 1$$

この制約条件あり最適化問題を，双形式ラグランジアン $\mathcal{L}(\alpha_1, \ldots, \alpha_N)$ で表す．ここで，α_n は N 個のラグランジュ乗数．ノイズを許容するハイパーパラメータ C を持つソフトマージン SVM は次の通り．

$$\mathcal{L} = \sum \alpha_n - \frac{1}{2}\sum\sum\ \alpha_n \alpha_m \ell^n \ell^m (\vec{x}^n \cdot \vec{x}^m)$$

として，

$$arg\,\max_{\alpha_n}\ \mathcal{L}(\alpha_1, \ldots, \alpha_N)$$

$$\textbf{s.t.}\ \ 0 \leq \alpha_n \leq C, \qquad \sum_{n=1}^{N} \alpha_n \ell^n = 0$$

分離超平面を特徴つけるパラメータは，ラグランジュ乗数から求めることができる．ただし，S はサポートベクタの集まり（図 1(a)）．

$$\vec{w} \ = \ \sum_{n \in S} \alpha_n \ell^n \vec{x}^n,$$

$$b \ = \ \frac{1}{|S|}\sum_{m \in S}(\ell^m \ - \ \sum_{n \in S} \alpha_n \ell^n (\vec{x}^n \cdot \vec{x}^m))$$

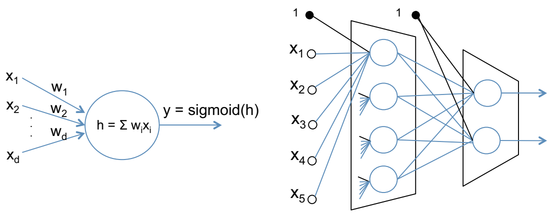
得られた関数 $y(\vec{x}) = \vec{w}^T \cdot \vec{x} + b$ を用いて，試験データセット D_T の正解率を調べれば良い．

4.1.2 分離超平面の近傍

サポートベクタマシンは図 1 からわかるように，分離超平面の近傍に位置するデータ点が計算結果に大きく影響する．フォローアップ入力として，このような限界データ点を追加する系統的な方法を考えれば良い．表 1 の Inclusive に相当する．

今，N 個のデータ点からなる訓練データセット $D^{(m)}$ に対して関数 $y^{(m)}$ が得られたとする．データ点を $X^j = \langle \vec{x}^j, +1 \rangle$ $(j = N+1, N+2)$ として，フォローアップ入力を $D^{(m+1)} = D^{(m)} \cup \{X^{N+1}, X^{N+2}\}$ とする．ただし，2 つのデータ点は分離超平面上に位置する（$y^{(m)}(\vec{x}^j) = 0$）．この $D^{(m+1)}$ に対して SVM を計算すると，$y^{(m)} = 0$ を満たすデータ点をサポートベクタとする新たな分離超平面 $y^{(m+1)}$ が求まる．関数 y はラグランジュ乗数に依存するので，$D^{(m+1)} = T(D^{(m)}, g(\{\alpha_n\}))$ の形になっていることがわかる．なお，限界データ点は多次元であり，方程式 $\vec{w}^{(m)} \cdot \vec{x}^j + b^{(m)} = 0$ の解として得られる．条件が不足しているので唯一に定まらない．文献 [6] では，その時点におけるサポートベクタの近傍に配置するというヒューリスティックスを用いた．

初期データセット $D^{(0)}$ が与えられた時，上記の変換 T を繰り返し適用すると，分離超平面 $y^{(0)} = 0$ 近傍にデータ点が集中する偏ったデータ分布のデータセットを順次得ることができる．今，$y^{(0)} = 0$ に関わるマージンを d_0 とする時，$D^{(m)}$ に対しては，$|d_0/2^m|$ となる．この過程を続けると，マージンがさらに小さくなり，ある時点で，微少量 ϵ_d に対して，$|d_0/2^m| \leq \epsilon_d$ を満たす．この時，分離超平面の両側に位置する 2 つのサポートベクタは，数値計算誤差の範囲で同一視される．これら 2 つは異なるラベルを持つので，これは不整合な状況に対応する．つまり，コーナーケース検査の入力となる．



(a) Perceptron　　(b) Two Layered Network

図 2 Neural Network

4.2 ニューラル・ネット

4.2.1 損失関数の最小化

ニューラル・ネットは，回帰や分類（識別）などの学習タスクに利用可能な一般的な枠組みである（たとえば，[3] の第 5 章).

†3 線形分離可能な問題設定．

図 2 (a) はニューラル・ネットの基本構成になるパセプロトンの模式図．一連の入力信号 x_i を受け取り，$y = \sigma(\sum_{i=1}^{d} w_i x_i)$ の計算結果を出力する．ここで，σ を活性化関数と呼ぶ．

本稿で考察する古典的なニューラル・ネットは図 2 (b) のように，パセプトロンを 2 層に重ねた構造を持つ．D 個の入力信号は M 個の中間層（隠れ層）に伝播され，その後，R 個の出力層につながる．今，h と r を活性化関数とする時，k 番目の出力信号 y_k は次のように表せる．

$$y_k(\boldsymbol{V}, \boldsymbol{W};\ \vec{x}) = r(\ \sum_{j=0}^{M}\ v_{kj} h(\sum_{i=0}^{D}\ w_{ji} x_i)\ )$$

ここで，$\vec{x}$ は D 次元ベクタ（$D{\times}1$ 行列）である．$\boldsymbol{W}$ は隠れ層への入力リンクの重み w_{ji} を要素とする $M{\times}D$ 行列，$\boldsymbol{V}$ は出力層への入力リンクの重み v_{kj} を要素とする $R{\times}M$ 行列，とする．行列の乗算を $\cdot$ で明示すると，$\vec{y}$ を R 次元の出力信号として，

$$\vec{y}(\boldsymbol{V}, \boldsymbol{W};\ \vec{x}) = r(\boldsymbol{V}{\cdot}h(\boldsymbol{W}\vec{x}))$$

と書ける．今，正解ラベルを $\vec{t^n}$ として，N 個の入力データ点 $\langle \vec{x^n},\ \vec{t^n} \rangle$ $(n = 1, \ldots,\ N)$, からなるデータセットが与えられたとする．誤差二乗和で定義する損失関数

$$E(\boldsymbol{V}, \boldsymbol{W};\ \{\langle \vec{x^n}, \vec{t^n} \rangle\}) = \frac{1}{2} \sum \| \vec{t^n} - \vec{y}(\boldsymbol{V}, \boldsymbol{W}; \vec{x^n}) \|^2$$

を使って，最適化問題として定式化できる．

$$arg \min_{\boldsymbol{V}, \boldsymbol{W}}\ E(\boldsymbol{V}, \boldsymbol{W};\ \{\langle \vec{x^n}, \vec{t^n} \rangle\})$$

これは，目的関数が非線形の非凸最適値探索問題であり，数値的に解く勾配法が基本的な方法になる．学習率と呼ぶハイパーパラメータ η $(>\ 0)$ を与えて，$\nabla E(\boldsymbol{V}^{old}, \boldsymbol{W}^{old})$ から更新値を求める繰り返しアルゴリズムである．実際は，処理性能向上の工夫として，逆伝播法や確率勾配法などを用いる．さらに，解への収束性等の向上を目的とする学習トリックを採用する．特に，入力正規化では，$\vec{x^n}$ のすべての属性 x_i^n を属性 i ごとに正規分布 $Norm(0, 1)$ に変換する方法を用いる．

4.2.2 手書き数字認識

データセット多様性の問題を考察する際には学習タスクを決める必要がある．ここでは，手書き数字認識学習に対する方法を，1 桁の手書き数字 1 文字と正解ラベルからなる MNIST データセットを対象として具体的に説明する．MNIST では，1 枚の手書き数字

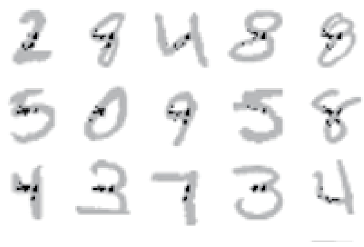


図 3　Multiplicative

は 28×28 の大きさを持つので，入力データは 784 次元となる．中間層は 50 次元とした．出力層は 10 次元である．各ピクセルはグレー画像で 0 から 255 までの値をとる．

まず，MNIST データセットの限界データ点を何にすれば良いかを考察する．手書き数字認識は，入力データのピクセル値（$\vec{x^n}$ の属性値）パターンから特定の数字と結論すること，パターンの分離ができないと他の数と誤る．ピクセルをうまく選んで，その値を変更すれば，パターンに影響がでる．そこで，表 1 の Additive や Multiplicative が良いメタモルフィック性になると期待できる．

本稿で提案するデータセット多様性に関わる変換関数は，次のような形式をとる．$D^{(m+1)} = T(D^{(m)}, G_o^N(\boldsymbol{V}^{(m)}, \boldsymbol{W}^{(m)}))$．次に示す関数 $top(N, o, \boldsymbol{V})$ を用いる．

今，$\boldsymbol{V}$ を 10×50 行列とする時，関数 $top(N, o, \boldsymbol{V})$ は，指定ラベル o（$o = 0, \ldots, 9$）を出力信号とする 50 個の行列要素 v_{oj}（$j = 1, \ldots, 50$）の中から，その重み値がトップ N の要素の添字 j からなる集合 H を返す．次に，H の要素をラベルとして $\boldsymbol{W}$ に同様の処理を施す．ここで，$h{\in}H$ をひとつ選ぶとすると，行列要素 w_{hi}（$i = 1, \ldots, 784$）の中から，その値がトップ N の要素の添字 i からなる集合を返す．$\bigcup_{h \in H} top(N, h, \boldsymbol{W})$ は指定ラベル o を持つ出力信号に影響を与えやすい入力信号の添字集まり U を表す．

関数 T は，上記の処理で得た添字の集まり U の要素について，データセット $D^{(m)}$ が含む $\vec{x^n}$ の u 属性値 x_u^n（$u{\in}U$）を更新する．例えば Multiplicative とし，$x_u^n{\times}F$ からなる $\vec{x^n}$ を $D^{(m+1)}$ の要素とする．

図 3 は，指定ラベルを 4 で選択する重みの数を 5

(つまり G_4^5) で，乗算係数を 4 とした場合を示す．手書き数字 4 の正解値に影響を与えると思われる入力属性の値を強めたもの．数字上に黒ドットとして見える．これを新たな $D^{(m+1)}$ として検査する．

実験によると，乗数係数を大きくすると黒ドットと他ピクセスの差が相対的に大きくなり，入力正規化によって全体的に不鮮明な画像になる．繰り返し行って，$D^{(m+k)}$ を得る場合を考える．繰り返しと共に，黒ドット数が増加・拡散し認識特徴が曖昧になり，その結果，解への収束性が悪くなる．

なお，上記では，ひとつのラベルを指定し，これを G_o^N と表したが，複数のラベルを指定してもよい．2 つを選ぶ ($G_{o1}^{N1} \cup G_{o2}^{N2}$) 等に拡張できる．

5 考察

統計的な機械学習では，学習性能・品質は与えたデータセットに大きく依存する．このことに着目した研究テーマがある．

能動学習 [4] は，学習パラメータ推定が効率よく進むようにデータ点を獲得する方法．学習モデルを獲得する訓練データセットは標本であって，想定する母分布に対して統計的な変動を示す．能動学習は，ユーザに問い合わせることで，想定通りの標本データ分布にしたがうデータ点を生成する．毒化攻撃 [2] は，訓練データセットに意図的に不正なデータ点を挿入し，適正な学習パラメータ推定を阻害しようとするもの．想定するデータ分布から逸脱したデータ点を生成する．学習タスク依存である．機械教師 [9] は，期待する学習パラメータ値を与えた時，この値への収束効率が最適となるような訓練データセットを生成する．訓練負荷を目的関数とする最小化問題として定式化される．

本稿では好ましい偏りのある多様なデータセットを系統的に生成する方法を論じた．データセット多様性は，上記の機械学習分野の問題とソフトウェア・テスティングに共通する見方といえる．

6 おわりに

一般に，ソフトウェア・テスティングでは，検査入力を系統的に生成する方法が不可欠である．求めた入力によって欠陥検出が必ず可能なわけではない．多様な入力の検査を行うことで，検査結果に対する信用レベルを高める．

本稿ではメタモルフィック・テスティングの拡張として，データセット多様性を考慮したフォローアップ入力生成の新しい方法を提案した．メタモルフィック関係の検査については紙面の都合で触れることができなかった．サポートベクターマシンについては文献 [6] に詳細記述がある．ニューラルネットの場合について，時系列分析を応用した振舞いオラクルと呼ぶ方法の有用性を実験で確認している．別稿で報告したい．

謝辞 実験に協力して下さった今井克則氏 ((株) グラッツ) に感謝する．

参 考 文 献

[1] P. Ammann and J.C. Knight : Data Diversity: An Approach to Software Fault Tolerance, *IEEE TC*, vol.37, no.4, pp.418-425, 1988.
[2] B. Biggio, B. Nelson, and P. Laskov : Poisoning Attacks against Support Vector Machines, In *Proc. 29th ICML*, 2012.
[3] C.M. Bishop : *Pattern Recognition and Machine Learning*, Springer-Verlag 2006.
[4] K. Brinker : Incorporating Diversity in Active Learning with Support Vector Machines, In *Proc. 20th ICML*, 2003.
[5] T.Y. Chen, S.C. Chung, and S.M. Yiu : Metamorphic Testing - A New Approach for Generating Next Test Cases, HKUST-CS98-01, The Hong Kong University of Science and Technology, 1998.
[6] S. Nakajima and H.N. Bui : Dataset Coverage for Testing Machine Learning Computer Programs, In *Proc. 23rd APSEC*, pp.297-304, 2016.
[7] E.J. Weyuker : On Testing Non-testable Programs, *Computer Journal*, 25 (4), pp.465-470, 1982.
[8] X. Xie, J.W.K. Ho, C. Murphy, G. Kaiser, B. Xu, and T.Y. Chen : Testing and Validating Machine Learning Classifiers by Metamorphic Testing, *J. Syst. Softw.*, 84(4), pp.544-558, 2011.
[9] X. Zhu : Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education, In *Proc. 29th AAAI*, pp.4083-4087, 2015.