情報論的学習理論テクニカルレポート 2009
Technical Report on Information-Based Induction Sciences 2009 (IBIS2009)

Virutual Concept Drift 環境における RBFNN のモデル選択
Model selection for RBFNN under Virtual Concept Drift Environments

山内康一郎*

Abstract: In this research, a model-selection criterion for RBFNN under virtual concept drift environments is proposed. Under such environments, the prior distribution of learning samples is changing over time so that online learning tasks usually cause catastrophic forgetting. Such environments are parts of *covariate shift*.
First of all, a statistical model of such environments is constructed. Then, we applied the learning strategies under covariate-shift using the statistical model. The method also provides the model selection criterion. Moreover, several strategies for reducing the computational complexity are also discussed.

Keywords: Virutal Concept Drift, 追記学習，共変量シフト，RBFNN, 汎化誤差，分布予測，t 分布, 情報量基準

1 まえがき

同時分布 $P(\boldsymbol{x}, y) = P(y|\boldsymbol{x})P(\boldsymbol{x})$ から発生する学習サンプル $(\boldsymbol{x}_b, y_b)$ $(b = 1, 2, \cdots)$ によって，入力ベクトル $\boldsymbol{x}$ と出力 y との間の条件付き分布 $P(y|\boldsymbol{x})$ を Radial Basis Function Neural Network(RBFNN) に学習させることを考える．これを online 学習で実現するには，独立かつ同一分布 (i.i.d) の $\boldsymbol{x}$ が与えられることを前提としなければならない．しかし，現実の環境では $P(\boldsymbol{x})$ 自体が時間と共に変化することが多く online 学習が Catastrophic forgetting を起こして失敗する．これをここでは Virtual Concept Drift 環境 [1] と呼ぶ．

この問題を克服するため多くの研究者が追記学習法を提案してきた [2–9]．これらのシステムは忘却を抑制しつつも，個々の学習サンプルを一度提示されるだけで学習が完了するという one-pass-learning を可能とする．これは，学習サンプル提示直後に運用期間に移ったネットワークが，いかなる入力が与えられても可能な限り誤りリスクを少なくするための学習戦略である．つまり，少なくとも既学習サンプルに似た入力に対しては確実に正解できるようにしておくことによって，誤りリスクを最小限に抑えようとするヒューリスティックな学習戦略である．これはすなわち，未来に与えられるサンプル分布が既学習サンプル分布を包含し，未知のサンプルを含む分布であることを仮定しており，既学習サンプルそのものとは異なる分布であることを仮定した学習戦略と言える．

一方，近年，学習サンプルの分布とテストサンプルの分布が異なる場合の学習戦略に関して深い議論がなされるようになってきている (e.g [10] [11])．このような環境のことを一般に共変量シフト (Covariate Shift) と呼び，テストサンプルに対する誤差を最小にする重みつき評価関数や，モデル選択基準も開発されている．もしこれらの知見を追記学習法に適用できれば，理論的に妥当な追記学習アルゴリズムとこれに適したモデル選択法の両方を構築できる可能性がある．

しかしながら，追記学習ではテストサンプル分布は未来に提示される新しいサンプルに相当するため，未知であると仮定せねばならない．残念ながら従来の研究にはこのテストサンプルの分布を予測する研究はほとんど見当たらず，これまでの研究成果をそのまま追記学習に適用することができない．

そこで筆者は，既に与えられたサンプルから，未来に提示されるサンプルの分布を予測する方法を考案し，これを使用した追記学習アルゴリズムの構築を試みた [12,13]．さらに，このような環境に適した RBFNN のモデル選択基準を与え，その性能を評価した [14]．本研究ではこれに加えて計算量の削減方法について考察を行った．

次節 2 ではここで想定する学習器について説明する．3 では追記学習環境のモデルとして未来に提示されるサンプルの予測分布の導出を試みる．4 では 3 で求めた予

*中部大学工学部 情報工学科, 〒 487-8501 愛知県春日井市松本町 1200, tel. 0568-51-9391, e-mail yamauchi@cs.chubu.ac.jp,
Chubu University Department of Information Science, 1200, Matsumoto-cho, Kasugai-shi, Aichi, 487-8501, JAPAN

測分布を用いたモデル選択付き追記学習アルゴリズムを示す．6 ではいくつかのベンチマークテストの結果を示し，7 でまとめる．

2　想定する学習システム

本研究では以下のように簡単化した追記学習システムを仮定する (図 .1)．すなわち，Radial Basis Function Neural Network(RBFNN) の横に提示された学習サンプル全てを保持するバッファが設けられてある．システムは学習サンプルの収集期間とリハーサル期間 (再学習期間) とを繰り返す．

これは，各々の学習期間において新しいサンプルのみならず過去の学習サンプルの再学習も行うタイプの追記学習法 [2,3,5–7,9] に共通する基本的な構造となっいる．

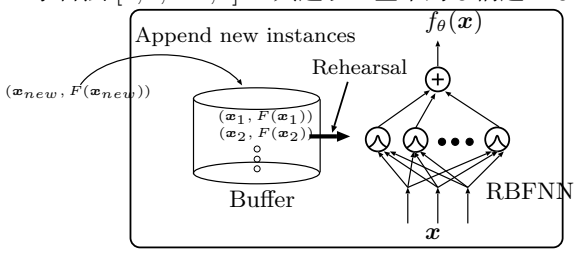

図 1: 本研究で仮定する学習システム

但し，ここでは簡単のためバッファのサイズに制限は無く，無限個貯められるものとする．またリハーサル期間中，RBF はバッファに貯められた全サンプルをオフライン学習するものとする．また RBF は毎度リセットされ学習と後述するモデル選択とを最初からやり直すものとする．このように新しいサンプルが与えられる度に学習をやり直すことから RBFNN 単体で見た場合には追記学習とはかけはなれているように見えるかもしれないが，バッファを含めた本学習システム全体に対しては，逐次的に学習データを与える形となっている．

$f_\theta(\boldsymbol{x})$ を RBFNN の出力値とすると，

$$f_\theta(\boldsymbol{x}) = \sum_{j=1}^{M} w_j \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{u}_j\|^2}{2v_j^2}\right), \qquad (1)$$

ここに M は隠れユニット (Kernel) の数である．

ここでの学習の目的は以下の評価関数を最小化することである．

$$E = \int (F(\boldsymbol{x}) - f_\theta(\boldsymbol{x}))^2 q(\boldsymbol{x}) d\boldsymbol{x}, \qquad (2)$$

ここに $F(\boldsymbol{x})$ は望ましい出力値を表し，$q(\boldsymbol{x})$ は真の入力分布を表している．$q(\boldsymbol{x})$ は学習サンプルの分布 $P(\boldsymbol{x})$ とは異なるものであることに注意する．

3　追記学習環境のモデル化

共変量シフト環境下での学習では以下のような重みつき誤差関数を使用する．

$$E = \sum_b W(\boldsymbol{x}_t) \left\{y_t - f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)\right\}^2 \qquad (3)$$

ここに $(\boldsymbol{x}_t, y_t)$ は各々の学習サンプルを表す．$W(\boldsymbol{x})$ は各々のサンプルの重みを表し，

$$W(\boldsymbol{x}) = \left(\frac{q(\boldsymbol{x})}{P(\boldsymbol{x})}\right)^\lambda \qquad (4)$$

である．ここに $P(\boldsymbol{x})$ は学習サンプル分布，$q(\boldsymbol{x})$ はテストサンプル分布を表す．$0 \le \lambda \le 1$ は平坦化パラメータである．

追記学習では $q(\boldsymbol{x})$ は将来提示されるサンプル全体の分布を表すものと仮定せねばならない．すなわち，$q(\boldsymbol{x})$ を $P(\boldsymbol{x})$ から予測する必要がある．これを如何にして行うかがここでの主要課題となる．

3.1　$q(\boldsymbol{x})$ の予測

今，N 個のサンプルが既に与えられ，バッファに格納されているものと仮定する．この少数のサンプルから考えうる $q(\boldsymbol{x})$ の候補はいくつも考えられる．そこで $q(\boldsymbol{x})$ をこの候補の平均分布 (以降 $\hat{q}(\boldsymbol{x})$ と記す) で近似するものとする．$\hat{q}(\boldsymbol{x})$ は以下のように表すことができる．

$$\hat{q}(\boldsymbol{x}) = \int P(\boldsymbol{x}|S) P(S|\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots \boldsymbol{x_N}) dS, \qquad (5)$$

ここに S は分布を表すパラメータベクトルを表す．ここで最も簡単な仮定として，$q(\boldsymbol{x})$ が正規分布であるとすると，$\hat{q}(\boldsymbol{x})$ は自由度 $N-1$ の *Student's-t* 分布で表せることが知られている [15]．すなわち，

$$\begin{aligned}\hat{q}(\boldsymbol{x}) \quad = \quad & \frac{\Gamma\left[(N-1+p)/2\right]}{((N-1)\pi)^{p/2}\Gamma\left[(N-1)/2\right]|\Sigma|^{1/2}} \times \\ & \left[1 + \frac{(\boldsymbol{x}-\boldsymbol{u})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{u})}{N-1}\right]^{-(N-1+p)/2} \end{aligned} \qquad (6)$$

ここに $p = dim(\boldsymbol{x})$, $\boldsymbol{u} = E[\boldsymbol{x}]$ そして Σ は分散共分散行列である．

ここで $P(\boldsymbol{x})$ は正規分布を仮定して，現在までに得られた N サンプルから ML 法で予測する．すなわち，$\hat{P}(\boldsymbol{x})$ を $P(\boldsymbol{x})$ の予測分布とすると，$\hat{P}(\boldsymbol{x})$ は $\mathcal{N}(\Sigma, \boldsymbol{u})$, $\Sigma = \frac{1}{N}\sum_{b=1}^{N}(\boldsymbol{x}_b - \boldsymbol{u})(\boldsymbol{x}_b - \boldsymbol{u})^T$, $\boldsymbol{u} = \frac{1}{N}\sum_{b=1}^{N}\boldsymbol{x}_b$ で表される．すなわち, $\hat{q}(\boldsymbol{x})/\hat{P}(\boldsymbol{x})$ は

$$\frac{\hat{q}(\boldsymbol{x})}{\hat{P}(\boldsymbol{x})} \quad = \quad \left(\frac{2}{N-1}\right)^{p/2} \frac{\Gamma[(N-1+p)/2]}{\Gamma[(N-1)/2]}$$

$$\times \quad \frac{\left[1+\frac{(\boldsymbol{x}-\boldsymbol{u})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{u})}{N-1}\right]^{-(N-1+p)/2}}{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{u})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{u})\right)}. \quad (7)$$

Student's-t 分布はサンプル数 N が大きくなるにつれて真の正規分布に近付いていくため，サンプル数 N が大きくなるにつれて $\hat{q}(\boldsymbol{x})/\hat{P}(\boldsymbol{x})$ は 1 に近付いていくことに注意されたい．

しかし経験上，現実の環境では $P(\boldsymbol{x})$ と $q(\boldsymbol{x})$ は複雑な形状を持っていることが普通であり，単一の正規分布、*Student's t* 分布のみで対処できるとは考え難い．

さらに，既に冒頭で述べたように，提示されるサンプルは i.i.d であるとも限らず，ある短時間内での学習サンプルの分布を観測した場合，その時の状況によって大きく異なった特定領域に集中的に提示されることが多い (例えばロボットのセンサーから得られるデータ等)．そこで，以下では，より現実の問題に適合する分布予測法を考える．

3.2 サンプル分布が変遷する場合の $q(\boldsymbol{x})$ の予測

その時々の状況 (以後 S_i $(i=1,2,\cdots)$ で表す) に強く依存したサンプルが集中的に現れ，且つ変遷する (図 2 参照) 場合を考える．各状況 S_i はそれまでの変遷履歴に応じて次の状態に遷移するが，時間が経つと再度同じ状況が現れるものと仮定する．

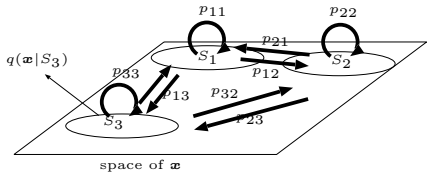

図 2: サンプル分布の遷移．(各々の状態は入力空間上の対応する位置で表現される)

すなわち，この状態遷移がマルコフ過程で表現されるものとし，かつエルゴード的であるとするならば，十分長い時間が経つと各状態 S_i となる確率はその初期状態に依存しない確率となり，$\hat{q}(\boldsymbol{x})$ は近似的に

$$\hat{q}(\boldsymbol{x}) \simeq \sum_i q(\boldsymbol{x}|S_i)p(S_i) \quad (8)$$

と表すことができる[1]．つまり，$\hat{q}(\boldsymbol{x})$ は混合分布として近似的に表せると予想される．同様に $\hat{P}(\boldsymbol{x})$ についても

$$\hat{P}(\boldsymbol{x}) \simeq \sum_i P(\boldsymbol{x}|S_i)p(S_i) \quad (9)$$

すなわち，

$$\frac{\hat{q}(\boldsymbol{x})}{\hat{P}(\boldsymbol{x})} = \frac{\sum_i q(\boldsymbol{x}|S_i)p(S_i)}{\sum_i P(\boldsymbol{x}|S_i)p(S_i)} \quad (10)$$

ここで、$P(\boldsymbol{x}|S_i)$ は正規分布で表されるものとし，$q(\boldsymbol{x}|S_i)$ は $P(\boldsymbol{x}|S_i)$ の中心位置と分散共分散行列を使って表現された *Student t* 分布であるとする．すると、次のように近似できる．

$$\frac{\hat{q}(\boldsymbol{x})}{P(\boldsymbol{x})} \simeq \frac{q(\boldsymbol{x}|S_j)p(S_j)}{P(\boldsymbol{x}|S_j)p(S_j)} = \frac{q(\boldsymbol{x}|S_j)}{P(\boldsymbol{x}|S_j)} \quad (11)$$

ただし，

$$j = \arg\max_i P(\boldsymbol{x}|S_i) \quad (12)$$

である．

各々の $p(\boldsymbol{x}|S_i)$ は正規分布とし，その中心位置と分散共分散行列は Expectation and Maximization (EM) アルゴリズム [16] を使用して決定した．またそのモデル数 (正規分布の数) は AIC [17] を使用して決定した．

$q(\boldsymbol{x}|S_i)$ は $p(\boldsymbol{x}|S_i)$ と同一の中心位置を持つ *Student-t* 分布である．したがって重みは各々のクラスターについて求めることとなる．すなわち，$W(\boldsymbol{x})$ は式 (11) より

$$W(\boldsymbol{x}) = \left\{ \left(\frac{2}{N_i-1}\right)^{p/2} \frac{\Gamma[(N_i+p-1)/2]}{\Gamma[(N_i-1)/2]} \times \frac{\left[1+\frac{(\boldsymbol{x}-\boldsymbol{u}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{u}_i)}{N_i-1}\right]^{-(N_i+p-1)/2}}{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{u}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{u}_i)\right)} \right\}^{\lambda}, \quad (13)$$

ここに

$$i = \arg\max_j \frac{1}{(2\pi)^{p/2}|\Sigma_j|^{1/2}} \times \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{u}_j)^T\Sigma_j^{-1}(\boldsymbol{x}-\boldsymbol{u}_j)}{2}\right). \quad (14)$$

ここに N_i, Σ_i, $\boldsymbol{u}_i$ はそれぞれ，第 i クラスタに属するサンプル数, 分散共分散行列，平均ベクトルである．

ただしこの手法は，既に与えられたサンプルのみを使用して，重み $W(\boldsymbol{x})$ を近似的に求めようとするものであり，サンプル数の少ない学習の初期段階では，当然の事ながらその後に与えられる全サンプルの分布を考慮した $W(\boldsymbol{x})$ を得ることは困難である．ここでは，既に与えられたサンプルの近傍に与えられるサンプルの分布を考慮した $W(\boldsymbol{x})$ である．

[1] 状態遷移する時間間隔はパターンが提示される時間間隔よりも十分に長いと仮定するが，特にこの事を明示的に数式には表現していない．これは S_i の状態遷移がエルゴード性を持つと仮定し，将来に渡って提示される全サンプル分布の平均として求めるため，不要となる．

4 学習アルゴリズム

3.2 で求めた重み $W_i(\boldsymbol{x})$ を使用して RBFNN の学習アルゴリズムを構築する．RBFNN は，リハーサル期間中，重みつき誤差関数式 (3) を最小化するように，バッファに保存された全てのサンプルを学習するものとする．ここでは様々な学習法が採用可能であるが，出来る限り正確な評価ができるように，中間ユニットと出力ユニットの間のコネクションは，Weighted Least Squre (WLS) で最適化する．中間ユニットの中心位置と分散は，Moody らが提案した手法 [18] に則って決めることにした．すなわち，この方法では k 個の中間ユニットの中心位置を k-means 法によって決定し，各ユニットの分散については、それに最も近いユニットの中心位置との距離によって決定する．また中間ユニットの数は Shimodaira によって提案された IC_w 4.2 によって決定する．

4.1 RBFNN の学習法

本研究では Moody らが提案した手法 [18] に若干の修正を加えた手法を採用する．すなわち，中間ユニットの中心位置と分散は重みつき fuzzy k-means 法を使って決定する．既に述べたように中間ユニットと出力ユニットの間のコネクションは WLS を使用して最適化する．

重みつき fuzzy k-means 法は, fuzzy k-means [19] 法を重みつきサンプルに適応させたもので，各クラスター中心位置 $\boldsymbol{u}_j$ は 1 ステップ前の中心位置できまるクラスターに属するサンプルの重みつき平均によって決定される．

$$\boldsymbol{u}_j^{(n+1)} := \sum_{b=1}^{N} \frac{W(\boldsymbol{x}_b)\boldsymbol{x}_b \exp(-\|\boldsymbol{x}_b - \boldsymbol{u}_j^{(n)}\|^2/c^2)}{\hat{c}_w \sum_{j'} \exp(-\|\boldsymbol{x}_b - \boldsymbol{u}_{j'}^{(n)}\|^2/c^2)}, \quad (15)$$

ここに $\hat{c}_w = \sum_{b=1}^{N} W(\boldsymbol{x}_b)$ であり，c は分散である．ただし各々の中心位置の初期位置は簡単のため B の最初の k サンプル, $\boldsymbol{x}_j$, に一致させた．この fuzzy k-means 法が収束した後，各 Hidden Unit の分散は

$$\sigma_j^2 = \kappa \; min_{j' \neq j} \|\boldsymbol{u}_j - \boldsymbol{u}_{j'}\|^2, \quad (16)$$

で決定する．ただし κ (≥ 1) はオーバラップファクター [20] である．

WLS は、中間ユニットと出力ユニットの間のコネクションの Eq(3) を最小化する最適解を解析的に求める．これをここではベクトル $\boldsymbol{w}_{ML} = (w_1, w_2, \cdots, w_M)^T$ で表す．ただし w_i は第 i 番めの中間ユニットと出力ユニットとの間の重みを表す．

$$\boldsymbol{w}_{ML} = (\boldsymbol{\Phi}\mathbf{W}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{W}\boldsymbol{F}, \quad (17)$$

ここに $\boldsymbol{F}$ は望ましい出力ベクトルであり、($\boldsymbol{F} = (F(\boldsymbol{x}_1), F(\boldsymbol{x}_2), \cdots, F(\boldsymbol{x}_N)))^T$) である。$\mathbf{W}$ は対角行列であり、その要素は $W_{bb} = W(\boldsymbol{x}_b)$ $(b = 1, 2, \cdots, N)$ で与えられる．$\boldsymbol{\Phi}$ は design matrix でありその各要素は $\Phi_{bj} = \exp(-\|\boldsymbol{x}_b - \boldsymbol{u}_j\|^2/(2\sigma_j^2))$ である．この学習法を使うことで，少なくとも中間ユニットと出力ユニットの間のコネクションについては必ず Eq.(3) を最小化する重みが得られるため、$W(\boldsymbol{x})$ の効果を評価し易くなる．

4.2 λ および中間ユニット数の最適化

汎化誤差を最小にするためには平坦化パラメータ λ と中間ユニット数 M をうまく決定する必要がある．これを行うために本研究では，Shimodaira(2000) [10] が提案した共変量シフト環境における情報量基準 IC_w を使用する．regression を扱うとすると IC_w は以下のように導出できる [10].

$$\begin{aligned} IC_w \quad := \quad & \sum_{b=1}^{N} \frac{\hat{q}(\boldsymbol{x}_b)}{\hat{P}(\boldsymbol{x}_b)} \left\{ \frac{\hat{\varepsilon}_b^2}{\hat{\sigma}^2} + \log(2\pi\hat{\sigma}^2) \right\} \quad (18) \\ + \quad & 2\sum_{b=1}^{N} \frac{\hat{q}(\boldsymbol{x}_b)}{\hat{P}(\boldsymbol{x}_b)} \left\{ \frac{\hat{\varepsilon}_b^2}{\hat{\sigma}^2}\hat{h}_b + \frac{W(\boldsymbol{x}_b)}{2\hat{c}_w} \left(\frac{\hat{\varepsilon}_b^2}{\hat{\sigma}^2} - 1 \right)^2 \right\}, \end{aligned}$$

ここに $\hat{\varepsilon}_b$ は入力 $\boldsymbol{x}_b$ に対する残差を表し，
$\hat{\sigma}^2 = \sum_{b=1}^{N} W(\boldsymbol{x}_b)\hat{\varepsilon}_b^2/\hat{c}_w$ である．ここに $\hat{c}_w = \sum_{b=1}^{N} W(\boldsymbol{x}_b)$ ．$\hat{h}_b$ $(b = 1, 2, \cdots, N)$ は以下で表される hat matrix の対角要素を表す．

$$\hat{\mathbf{h}} = \boldsymbol{\Phi}(\boldsymbol{\Phi}^T\mathbf{W}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{W}. \quad (19)$$

すなわち IC_w を最小にする組合せ (λ_*, M_*) を探せば良い．本研究では λ と M の組合せを予め幾つか用意し，その中で IC_w が最小となる組合せを解とした．

5 計算量削減法の検討

本手法において大きな計算量が必要となるのは，モデル選択の部分である．モデル選択は，$P(\boldsymbol{x})$ の予測のための混合分布の数，RBFNN の中間ユニット数およびパラメータ λ の決定のために実行される．

モデル選択では一般に複数の解候補を用意し，それらの中で最も情報量基準 (AIC/ IC_w) の小さい解候補を選択する．このとき横軸にパラメータ数，縦軸に情報量基準の値を取ってプロットすると多くの場合 V 字型のグラフとなる．そのため，混合分布のモデル数もしくは RBFNN の中間ユニット数を 1 個づつ増加させて学習させ，AIC もしくは IC_w の値が増加に転じた時点でモデルの生成を止め，ひとつ前のモデルを解とするようにすれば，大幅に計算量の削減が期待できる．

しかしながら，実際には情報量基準の値は V 字形状になるとは限らず最適なモデルが選択できない可能性もある．これを確認するため後述の実験では，まず混合分布のモデル選択に関してこのように簡略化した場合の性能の違いを検証した．

6 計算機実験

人工データセットとベンチマークデータセットを使って提案システムを評価した．本学習アルゴリズムは新しいサンプルの収集期間 (*recording* phase) とリハーサル期間 (*rehearsal* phase) とを繰り返すが，本手法の新規点である分布予測に基づく重みつき誤差関数の効果を調査するには，1 セットの *recording*, *rehearsal* phase のみで十分である．そこで，*recording* phase に提示するサンプルをランダムに変えて性能を評価した．なお，表現の簡素化のため重みつき誤差関数を使って学習を行うモデルを” WRBFNN”, 重みを付けない従来の誤差関数を使って学習を行うモデルを”org-RBFNN” と表記することにする．org-RBFNN は従来の追記学習法 [2–7, 9] に相当する．また org-RBFNN は $\lambda = 0$ とした WRBFNN と同一であることに注意する．

6.1 1 次元人工データに対する $W(\boldsymbol{x})$

$W(\boldsymbol{x})$ がどのように設定されるかを見るために 1 次元の人工データで動作を確認した．

$$\begin{aligned}(x, y) &= (x, 1.5) \\ x &\sim \frac{1}{2}\mathcal{N}(-20, 2) + \frac{1}{2}\mathcal{N}(20, 2) \qquad (20)\end{aligned}$$

ここから 50 個のサンプルを生成した．さらに Eq. (13) の効果を明示的に示すため，$\mathcal{N}(10, 5)$ から 3 つのデータを生成して孤立点として加えた．

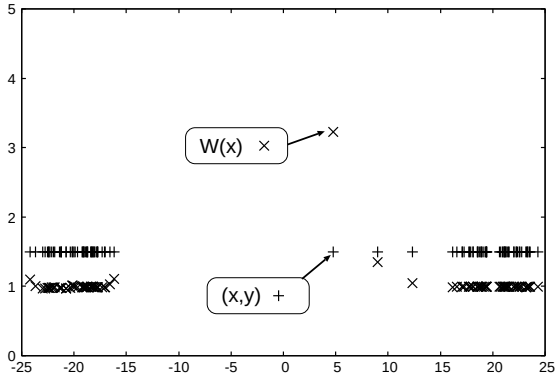

図 3: $W(\boldsymbol{x})$ の例

この図から分布の端の方に位置する x 程，重みが増加していることが分かる．孤立点の重みはさらに大きくなっている．すなわちこのような孤立点に対して，RBFNN がより強く学習することを意味している．

6.2 ベンチマークテスト

平坦化パラメータ λ と中間ユニット数 M の最適値 (λ^*, M^*) を探し，この最適値における WRBFNN のパフォーマンスと同じ中間ユニット数 M^* における org-RBFNN のパフォーマンスを比較した．本稿では UCI-machine learning repository に収録されてある cpu-performance データセットに対するパフォーマンスのみ示す[2]．5 で述べた計算量の削減法を考察するためモデル選択の方法に応じてパフォーマンスの比較を併せて行った．このため，以降，簡略化されたモデル選択法を採用するものを”-Quick” を付けて示すものとする．すなわち Mixture of Gaussian を使用した分布推定に関するモデル選択については，$AIC - org$ と記し，その簡略化されたものを $AIC - Quick$ と記すことにする．

本来ならば recording-, rehearsal-phase を交互に何度も繰り返す学習法ではあるが，性能を評価するには 1 回の recording, rehearsal-phase だけで十分である．

データセットからは，その一部の 100 個データをランダムに選択したものを 50 セット用意した．それぞれの 100 個のデータは recording phase においてバッファ B に貯められるものとみなす．

分布推定にあたっては分布の分散共分散行列を簡単のため対角要素のみ扱い，パラメータの推定を行った．λ を探す際の刻幅は 0.1 とした．また，中間ユニット数の上限を 20 個としてある．また評価用のテストデータは全データセットとした．ただし評価結果に関してはデータセットの選びかたによって誤差が大きく変動するため，

$(x, y) = (MSE_{WRBFNN}, MSE_{org-RBFNN})$

をプロットして評価する．ここに

$$MSE_* \equiv \frac{1}{N_{total}} \sum_{b=1}^{N_{total}} (F[\boldsymbol{x}_b] - f_{\theta_*}(\boldsymbol{x}_b))^2, \qquad (21)$$

である．すなわちこのようにして打たれた点が直線 $y = x$ よりも上に分布しているならば，WRBFNN のパフォーマンスが org-RBFNN のそれよりも良いことを意味する．但し，学習が不安定に陥るのを防ぐため $W(\boldsymbol{x})$ の値が 10 以上にならないように制限を加えた．

図 4 は各学習データセットに対する

$(x, y) = (MSE_{WRBFNN}, MSE_{org-RBFNN})$

をプロットし，したものである．本手法においては各々の学習データセットにおいて最適な λ と中間ユニット数 M を探索するが，場合によっては $\lambda = 0$ すなわち WRBFNN でありながら org-RBFNN と同一の学習法

[2]他のデータセットに対するパフォーマンスについては文献 [14] を参照されたい．ただし [14] では分布推定に関するモデル選択は簡略化されたものを使用している．

を選択することがある．つまり，この場合はポイントが $y = x$ 上に配置される．しかし $\lambda > 0$ を選択した場合にその効果があればパフォーマンスのポイントが $y = x$ の上側に配置されるが悪ければ下側に配置される．

この結果より cpu-performance に関してはパフォーマンスのポイントが $y = x$ よりも上に配置される頻度が高いため，本手法の効果が得られていると考えられる．

また，簡略化されたモデル選択アルゴリズムを使用した”AIC-Quick” とオリジナルのモデル選択アルゴリズムを使用した”AIC-org” を比較すると，概ね結果が重なっており，計算量を抑えてもパフォーマンスに与える影響が少ないことを示唆している．

7　まとめと考察

本研究では，追記学習が共変量シフトの一つであることを指摘し，これまでに研究されてきた共変量シフトに対処する学習戦略を組み込んだ追記学習法の構築を目指した．これを実現するために，既に与えられたサンプルからこの先将来に渡って提示されるサンプルの分布を予測する手法を考案し，これを使った重み付き誤差関数を定義した．

しかしながら，既に得られたサンプルから遠い将来に渡って提示されるサンプルを正確に予測することは不可能である．提案手法では，近い将来，既に与えられたサンプルの近傍に提示されるサンプルの分布を予測するものとなっている．これは次の意味において妥当である．すなわち，RBFNN を使用する場合，学習後のネットワークが対処できる入力領域 (定義域) は既に与えられた学習サンプルの近傍に限られる．したがって，既学習サンプルから遠く離れたサンプルに対しては，いかなる努力も果を結ばないと考えられるため，ここでの予測分布も既学習サンプルの近傍において有効に働けば良い．

本研究ではさらにこれに合う RBFNN の中間ユニット数を決定する手法も与え，常に与えられたデータセットにフィットし、且つ将来与えられるデータにもフィットするであろう RBFNN を構築することが出来る．また実験結果よりモデル選択アルゴリズムを簡略化することにより計算量を低く抑えることが出来ることも示した．

本稿での詳述は避けたが，複数のベンチマークテストも行っており，提案システムの効果は学習データセットに大きく依存することが分かっている [14]．つまり本手法において 3.2 で予測する Virtual Concept Drift 環境にフィットするデータであれば効果を発揮するものの，それ以外のデータ、すなわち学習データとデータセット全体の分布とが一致する場合などでは効果を発揮しない．

だが Virtual Concept Drift 環境は現実の学習環境においては頻繁に発生する．例えば太陽電池の照度とパネル表面温度に対する最大電力点の分布は太陽の傾きと共に時々刻々と変化する．さらに学校で毎日生徒に教える事柄に付いても，最初から全体を網羅する内容では無くそれぞれの項目を順番に教えていくため，Virtual Concept Drift と言える．したがってこのような現実の学習環境において本手法が必要とされると考える．

参考文献

[1] A. Tsymbal. The problem of concept drift: definitions and related work. Technical Report TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, 2004.

[2] Takao Yoneda, Masashi Yamanaka, and Yukinori Kakazu. Study on optimization of grinding conditions using neural networks – a method of additional learning –. *Journal of the Japan Society of Precision Engineering/Seimitsu kogakukaishi*, 58(10):1707–1712, October 1992.

[3] Hiroshi Yamakawa, Daiki Masumoto, Takashi Kimoto, and Shigemi Nagata. Active data selection and subsequent revision for sequential learning with neural networks. *World congress of neural networks (WCNN'94)*, 3:661–666, 1994.

[4] Stefan Schaal and Christopher G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084, November 1998.

[5] Koichiro Yamauchi, Nobuhiko Yamaguchi, and Naohiro Ishii. Incremental learning methods with retrieving interfered patterns. *IEEE transactions on neural networks*, 10(6):1351–1365, November 1999.

[6] Robert M. French. Pseudo-recurrent connectionist networks: An approach to the “sensitivity stability” dilemma. *Connection Science*, 9(4):353–379, 1997.

[7] Bernard Ans and Stephane Roussert. Neural networks with a self-refreshing memory: knowledge transfer in sequential learning tasks without catastrophic forgetting. *Connection Science*, 12(1):1–19, 2000.

[8] Nikola Kasabov. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 31(6):902–918, December 2001.

[9] Seiichi Ozawa, Soon Lee Toh, Shigeo Abe, Shaoning Pang, and Nikola Kasabov. Incremental learning of feature space and classifier for face recognition. *Neural Networks*, 18:575–584, 2005.

[10] Shimodaira Hidetoshi. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[11] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von B ¨unau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Twenty-First Annual Conference on Neural*

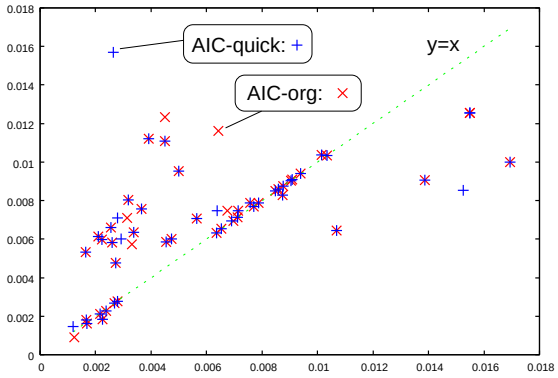

図 4: cpu-performance に対するパフォーマンス (AIC-org と AIC-quick を同時にプロットしてある)

Information Processing Systems (NIPS2007), December 2007.

[12] Koichiro Yamauchi. Covariate shift and incremental learning. In *Advances in Neuro-Information Processing 15th International Conference, ICONIP 2008, Auckland, New Zealand, November 25-28, 2008, Revised Selected Papers, Part I*, pages 1154–1162, November 2008.

[13] 山内 康一郎. 共変量シフトと追記学習. Technical Report NC2008-142, 電子情報通信学会技術報告, 3 月 2009.

[14] Koichiro Yamauchi. Optimal incremental learning under covariate shift. *Memetic Computing*, page Accepted, 2009.

[15] 繁桝 算男. **ベイズ統計入門**. 東京大学出版会, 1985.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B 39(1):1–38, 1977.

[17] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, December 1974.

[18] J. Moody and C. J. Darken. Fast learning in neural networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[19] J.C. Bezdek. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:1–8, 1980.

[20] John Platt. A resource allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.