Random Forest を用いた事例型追加学習

三品 陽平† 村田 隆英† 山内 悠嗣† 山下 隆義† 藤吉 弘亘†
†中部大学
E-mail: hf@cs.chubu.ac.jp

Abstract

統計的学習手法により構築した識別器は，学習サンプルと実利用シーンサンプルが異なると識別性能が低下するという問題がある．実利用シーンに対する性能を向上させるには，実利用シーンのサンプルを用いた追加学習が有効である．追加学習では，誤識別サンプルを用いて識別器の一部を追加更新する．しかし，追加学習の際に，学習サンプルに対する汎化性能が低下する恐れがある．本研究では，容易に実利用シーンへの適応が可能な追加学習を Random Forest の枠組みにおいて実現する．Random Forest の木構造を利用することで，実利用シーンに対して適応しながらも，学習サンプルに対する汎化性能への影響を最小限に抑制することができる．マルチクラス問題であるナンバープレートの陸支コードを対象とした評価実験と，2 クラスの分類問題である人検出を対象とした評価実験により，Random Forest を用いた事例型追加学習の有効性を確認した．

1 はじめに

統計的学習手法を用いた物体認識は，学習データセットを用いて事前に識別器を構築する．このとき，汎用的な学習用データセットを用いて構築した識別器は，外乱等の影響を受けると実利用シーンのサンプルを正しく識別できないことがある．これは，学習用サンプルと識別用サンプルから得られる特徴量の分布が異なるためである．実利用シーンでの誤識別を抑制する方法として，実利用シーンから収集したサンプルを用いて，再学習するアプローチと追加学習するアプローチが考えられる．

識別器の再学習では，実利用シーンに最適化された識別器を構築できる一方，実利用シーンのサンプルを大量に収集する必要がある．また，識別器を構築するためには膨大な時間が必要であり，シーンごとに識別器を再学習するには非常に高い計算コストが必要になる．一方，追加学習はシーンにおける誤識別サンプルを用いて識別器の一部を更新するため，少ない計算コストで最適化できる．しかし，追加学習では実利用シーンに過学習するために学習サンプルに対する識別性能が低下する恐れがある．

学習に小規模なサンプル集合を学習に用いることで計算量の低コスト化をはかる手法として能動学習 [1] がある．能動学習は，学習に寄与しそうな少数のサンプルを用いて学習する．能動学習におけるサンプルの選択は，一般的に識別境界付近に存在し，識別が難しいサンプルを曖昧なサンプルとして選択する．能動学習の代表的な手法として Margin Sampling[2] や，Entropy[3], Vote Entropy[4] 等がある．能動学習により誤識別サンプルを少数選択し，学習サンプルに加えることでサンプル収集に対する人的コストの削減も期待できる．

そこで，本研究では低計算コストかつ学習サンプルに対する識別性能を低下させない追加学習法を Random Forest の枠組みを用いて実現する．Random Forest は，木構造を利用しているため学習サンプルに対する汎化性能を維持しながら高速に実利用シーンへの適応が可能である．決定木内の誤識別サンプルが到達した末端ノード以降のみを更新することで，他のノードに影響を与えない．そのため，学習サンプルに対する汎化性能が維持することができ，誤識別サンプルが到達した末端ノードのみを更新することで高速に追加学習できる．また，能動学習により識別が難しいような誤識別サンプルのみを自動収集することで，人的コストを削減できるという利点がある．

2 Random Forest

本章では，Random Forest について述べる．Random Forest は，複数の決定木構造を持つマルチクラス識別器を構築する学習アルゴリズムである．Random Forest のアルゴリズムの特長は，Bagging[6] と同様にブートストラップを取り入れ過学習を防ぐ点，Random Feature Selection[7] を取り入れ特徴ベクトルの次元数が大きくても高速に学習が可能な点である．このようなメリットを持つため，コンピュータビジョンの分野でも，セマンティックセグメンテーション [8]，文字認識 [9]，物体認識 [10][11]，人体姿勢推定 [12] で用いられている．

2.1 学習

Random Forest は，学習サンプルからサブセットを作成し，複数の決定木構造を持つ識別器を構築する．各決定木は，分岐ノードと末端ノードにより構成され，分岐ノードを繰り返し作成し，一定の基準により分岐が不可能になった際に末端ノードを作成する．木の数 T, 木の最大の深さ D とする．

学習サンプル集合 $\mathcal{I} = x_1, y_1\}, \ldots, \{x_N, y_N\}, x_i \in \mathcal{X}, y_i \in \{1, 2, \ldots, C\}$ を入力し，サブセットを作成する．x は学習サンプルの特徴量ををを表し，y はクラスラベルを，c はクラス数を表す．サブセットは学習サンプル $\mathcal{I}$ からサンプルの重複を許容してランダムに選択する．サブセットの一つを用いて決定木を構築する．決定木の分岐ノードはある特徴量 f_k としきい値 τ_h を用いて左もしくは右へサンプルを分岐させる分岐関数が保存されている．分岐関数は，特徴選択回数 K としきい値選択回数 H から，候補をランダムに $K \times H$ 個選択し，情報利得 ΔE が最も高い候補を分岐関数として用いる．ある分岐ノード n にたどり着いたサンプル集合 $\mathcal{I}_n$ を分岐関数の候補 f_k と τ_h により $\mathcal{I}_l$ と $\mathcal{I}_r$ に分割する．この $\mathcal{I}_l$ と $\mathcal{I}_r$ を用いて，式 (1) により情報利得 ΔE を算出する．情報利得は，現在のノードのエントロピーから子ノードのエントロピーの和を引いたものであり，分岐関数によりどの程度情報量が減少したかを表している．子ノードのエントロピーが小さくなると情報利得は大きくなり，クラスをよく分割する分岐関数と表される．

$$\Delta E = E(\mathcal{I}_n) - \frac{|\mathcal{I}_l|}{|\mathcal{I}_n|} E(\mathcal{I}_l) - \frac{|\mathcal{I}_r|}{|\mathcal{I}_n|} E(\mathcal{I}_r) \qquad (1)$$

ここで，関数 $E(\mathcal{I})$ は情報エントロピーを表し式 (2) により算出される．

$$E(\mathcal{I}) = -\sum_{i=1}^{n} p(c_i) \log p(c_i) \qquad (2)$$

$p(c_i)$ はクラス c_i の確率を表しており，学習サンプルの教師信号の出現頻度により求められる．これらの処理を繰り返すことによりサンプル集合を分割し，情報利得が 0 になった場合や最大の深さ D に達した場合に末端ノード l を作成し，到達したサンプル集合から各クラスの出現確率 $P(c|l)$ を計算する．このように，各決定木が構築される．

2.2 識別

Random Forest の識別のアルゴリズムについて説明する．未知入力サンプル x をすべての決定木に入力し，分岐関数により左右に分岐させ決定木をトラバーサルする．そして，たどり着いた末端ノードに保存されているクラス確率 $P_t(c|x)$ を出力する．式 (3) に示すように，すべての決定木の出力の平均を算出する．

$$P(c|x) = \frac{1}{T} \sum_{t=1}^{T} P_t(c|x). \qquad (3)$$

最終出力 $\hat{y}$ は，式 (4) により最も確率の高いクラスとする．

$$\hat{y} = \arg\max_{c} P(c|x). \qquad (4)$$

3 提案手法

提案手法は，まず学習サンプルと誤識別サンプルを事前に構築した Random Forest の全ての決定木に入力する．その後，全ての決定木を走査し，誤識別サンプルのクラス確率が低い末端ノードに子ノードを追加していくことで追加学習を実現する．追加するノードの分岐関数には，特徴選択型と事例型の 2 つを提案する．

3.1 追加学習のフレームワーク

提案手法の追加学習のフレームワークを図 1 に示す．まず，学習用サンプルを用いて識別器を構築する．構築した識別器を用いて実利用シーンのサンプルを識別する．このとき，図 1 に示すような誤識別するサンプルを蓄積し，正しいラベルを与え，識別器を追加学習する．

図 2 に決定木の更新方法を，**Algorithm** 1 に Random Forest における追加学習アルゴリズムを示す．誤識別サンプルが到達した末端ノードに分岐関数を追加することで，誤識別サンプルと学習用サンプルを分離する．誤識別サンプルが到達した場合でも，誤識別サンプルクラスの出現確率が高い場合は，末端ノードを更新しない．誤識別サンプルを蓄積する方法は，目視により判別する方法と Vote Entropy を用いた評価指標により判別する方法を提案する．

Algorithm 1 Random Forest の追加学習アルゴリズム

Require: 学習サンプル $\{x_1, y_1\}, \ldots, \{x_N, y_N\}$;
$x_i \in \mathcal{X}, y_i \in \{1, 2, \ldots, M\}$
誤識別サンプル $\{x'_1, y'_1, \hat{y}_1\}, \ldots, \{x'_K, y'_K, \hat{y}_K\}$;
$x'_i \in \mathcal{X}, y'_i \in \{1, 2, \ldots, M\}$

Run:
すべてのサンプルをすべての決定木に入力.
for $t = 1 : T$ **do**
決定木を走査.
if $y' != \hat{y}$ **then**
末端ノードを更新する.
$\Delta E_{max} \Leftarrow -\infty$.
for $t = 1 : K_l$ **do**
誤識別サンプルからテンプレートを選択.
ユークリッド距離 D と τ_h によりサンプル集合 $\mathcal{I}_n$ を $\mathcal{I}_l$ と $\mathcal{I}_r$ に分割.
情報利得 ΔE を算出:
$\Delta E = E(\mathcal{I}_n) - \frac{|\mathcal{I}_l|}{|\mathcal{I}_n|} E(\mathcal{I}_l) - \frac{|\mathcal{I}_r|}{|\mathcal{I}_n|} E(\mathcal{I}_r)$.
if $\Delta E > \Delta E_{max}$ **then**
$\Delta E_{max} \Leftarrow \Delta E$
end if
end for
if $\Delta E_{max} = 0$ **then**
末端ノードにクラス確率 $P(c|l)$ を保存.
else
分岐を続ける.
end if
end if

end for

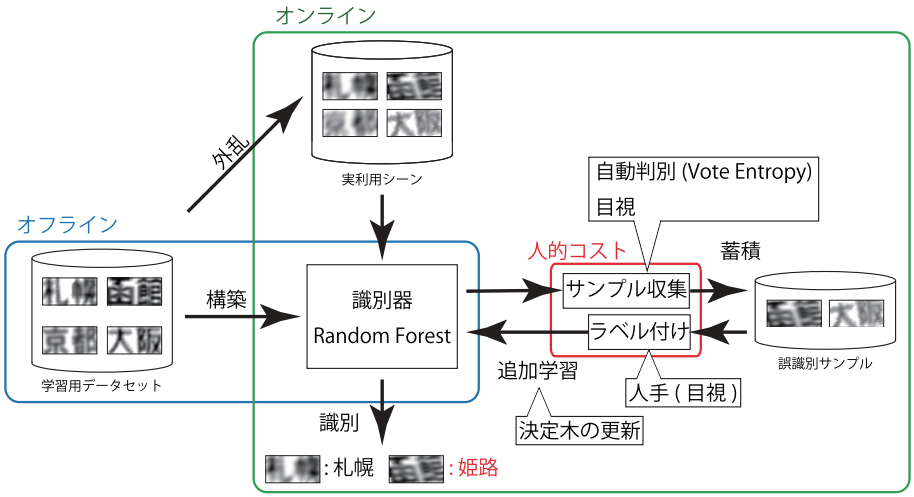

図 1　提案手法のフレームワーク

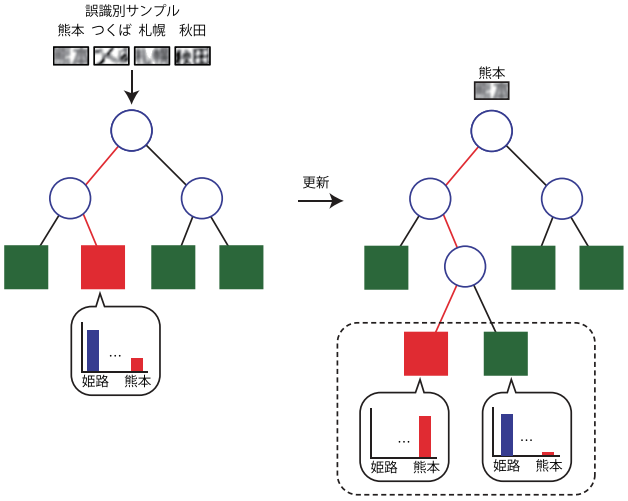

図 2　追加学習における決定木の更新

3.2　誤識別サンプルの自動収集とラベル付け

追加学習をするためには，実利用シーンにおいて誤識別サンプルを収集する必要がある．しかし，実利用サンプルにはラベルが与えられていないため，識別結果を目視によって確認する必要がある．目視によるサンプル収集には高い人的コストが必要であるため，本研究では Vote Entropy[4] によりサンプルを自動収集する．Vote Entropy は，アンサンブル識別器における識別結果の信頼性を評価することができ，Vote Entropy は式 (5) により求められる．

$$VE(x) = -\sum_{m=1}^{M} \frac{V(y_m)}{T} log \frac{V(y_m)}{T} \qquad (5)$$

ここで，T は決定木の数，$V(y_m)$ はラベル y_m と予測した決定木の数を表している．そのため，Random Forest としての解釈が曖昧なサンプルは誤識別している可能性が高いと考え，Vote Entropy が高いサンプルを自動収集し，目視で正しいラベルを付けたサンプルを追加学習に使用する．

3.3　特徴選択型の分岐関数

特徴選択型の分岐関数は，参照する特徴次元とそのしきい値をランダム選択した候補の中から情報利得を基に選択する．図 3 に特徴選択型の分岐関数例を示す．特徴量 $F(\cdot)$ には Haar-like フィルタを用いる．分岐関数は Haar-like フィルタのフィルタパターン，サイズ，位置を選択する．また，分岐関数を式 (6) に示す．

$$\begin{cases} \text{left} & \mathrm{F}(x) < \tau \\ \text{right} & \mathrm{F}(x) >= \tau \end{cases} \qquad (6)$$

ここで，$\mathrm{F}(x)$ は入力サンプルから Haar-like フィルタにより特徴抽出した値を，τ はしきい値を表す．

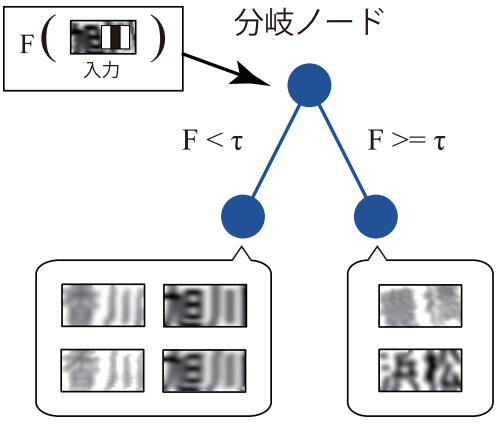

図 3　特徴選択型の分岐関数例

3.4　事例型の分岐関数

分岐関数に事例を用いる分岐方法 [13] が提案されている．文献 [13] では，人検出の問題において，事例型の分岐関数を手法に用いている．本稿における事例は，誤識別サンプルより選択されたテンプレートを指す．図

4 に事例型の分岐関数例を示す．事例を用いた分岐関数は，図 5 に示すノード A のように，ある誤識別サンプルをテンプレートとして，同じ末端ノードの学習サンプルと誤識別サンプルとの距離を計算し，最も距離の近いクラスの異なるサンプルとの中間点にしきい値を決定し分岐する．ノード B のように，追加学習後に誤識別サンプルが存在する場合はさらに追加学習を行う．距離計算には，ユークリッド距離を使用する．テンプレートとしきい値の組み合わせを情報利得を基に選択する．事例型の分岐関数を式 (7) に示す．

$$\begin{cases} \text{left} & \mathrm{D}(x_i, x_T) < \tau \\ \text{right} & \mathrm{D}(x_i, x_T) >= \tau \end{cases} \tag{7}$$

ここで，D はテンプレートと入力サンプルの距離を，τ はしきい値を表す．また，事例型の分岐関数のしきい値は，式 (8) によって決定する．

$$\tau = \frac{1}{2} \min_{\delta(y_i, y_T)=0} d_i \tag{8}$$

ここで，d_i はテンプレートとクラスの異なる学習サンプルの中で最も距離の近いサンプルを，T はテンプレートを表す．

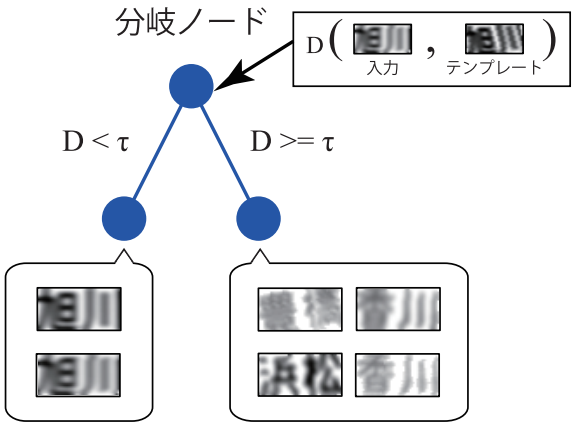

図 4　分岐関数例

このような分岐関数を用いて誤識別した末端ノードのみを更新することで，学習サンプルに対する識別性能を維持したまま誤識別サンプルに適応する追加学習を実現できる．追加学習において，実利用シーンの誤識別サンプルは汎用データセットの例外的なサンプルが多く，このような場合に事例型が有効に働くと考えられる．

4　評価実験

提案手法の有効性を評価するために評価実験をする．評価実験は 2 つのデータセットを用いて評価する．1 つ目は，マルチクラス問題であるナンバープレートの陸支コード認識を対象とする．2 つ目は，2 クラス問題である人検出を対象とする．

4.1　ナンバープレートの陸支コード認識実験

陸支コードは，居住地の運輸支局・自動車検査登録事務所において使用する自動車を登録することで発行される．ナンバープレートは，陸支コード，分類番号，用途コード，一連コードにより構成されている．今回は，陸支コードを認識対象とする．評価実験では，目視による誤識別のサンプル収集と VoteEntropy による誤識別サンプルの自動収集を評価し，k-NN(k=3)，Random Forest の再学習，提案手法を比較する．提案手法である追加学習は，特徴選択型の分岐関数と事例型の分岐関数を用いて追加する手法を比較する．学習パラメータは，木の数は 25，木の深さは 20，特徴次元選択回数は 100，しきい値選択回数は 200，サブセットサイズを 75%とする．

4.1.1　データセット

図 6 にデータセットの陸支コードの一例を示す．クラス数は 105 クラスあり，本実験に使用するデータは，各クラスにつき学習用に約 3,000 枚，追加学習用に約 500 枚，評価用に約 500 枚用いる．画像は，照明変化，幾何変化をランダムに与えた自動生成画像を含める．また，学習用と検証用，評価用には同じ画像は含まれない．ナンバープレートや文字の切り出し問題については，本実験では扱わない．そのため，切り出した画像をサイズや輝度が等しくなるように正規化処理をした画像を入力とする．

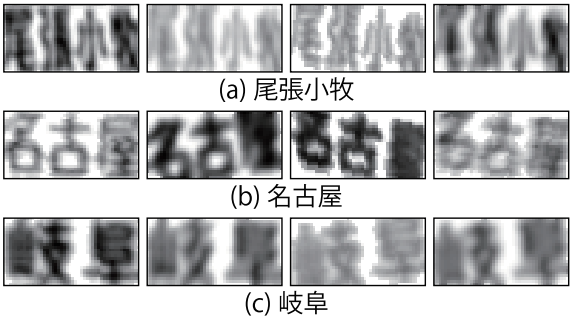

図 6　陸支コードのデータセット例

4.1.2　特徴量

本実験には，特徴選択型の分岐関数として Haar-like フィルタ [14] を用いる．図 7 に Haar-like フィルタのパターンを示す．今回は図に示す 5 種類を用いた．

図 7　Haar-like フィルタ

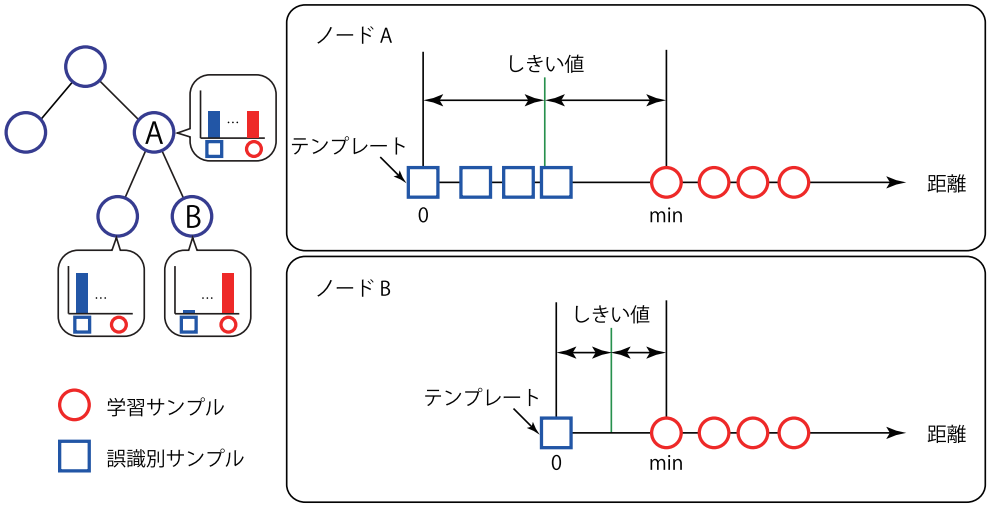

図 5　しきい値の選択方法

4.1.3　実験結果

実験結果を以下に示す．

目視にによる誤識別サンプルの収集

目視によって収集した誤識別サンプル 4841 枚からランダムサンプリングし，追加するサンプルの割合を 1%〜10%に変化させた場合の F 値を図 8 に示す．提案手法は，追加率 2%において 99%以上の識別性能が得られた．

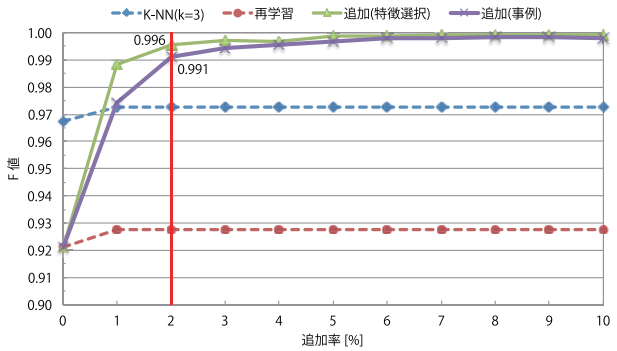

図 8　追加学習による識別性能の変化

Vote Entropy による誤識別サンプルの自動収集

目視によるサンプル収集と同様に，Random Forest を再学習する方法と追加学習 (特徴選択型，事例型) する方法を比較する．図 9 に F 値を示す．グラフより，図 8 に示すように目視によってサンプルを収集した方法と，Vote Entropy により誤識別サンプルを自動収集する方法を比較する両手法共に追加学習後の識別性能は追加率 2%において 98%以上と，目視によるサンプル収集と大きな差はなく，追加学習コストの削減に貢献している．

図 10 に，追加学習後の Confusion Matrix を示す．Confusion Matrix は，行方向に入力クラスを示し，列

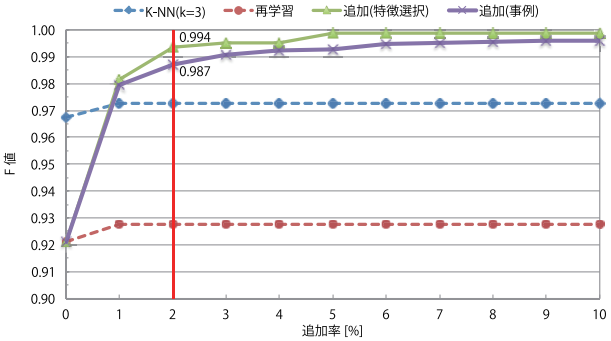

図 9　自動収集したサンプルの追加学習結果

方向に出力結果を示している．確率が高いほど赤くなり，低いほど青くなる．そのため，対角成分が赤いほど識別精度が高いことを示す．追加学習前では，誤識別しているクラスが存在しているが，追加学習後はほぼすべてのクラスにおいて 100%正解している．

図 11 に各手法の追加学習時間を示す．追加率 2%において事例型の学習時間は 34 秒であり，再学習の 1/4270，特徴選択型の 1/19 と高速な追加学習を実現した．

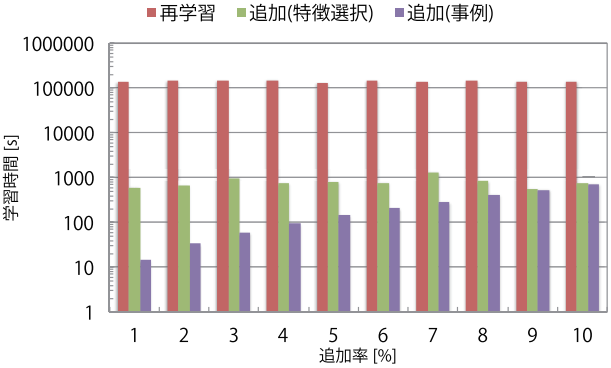

図 11　追加学習時間

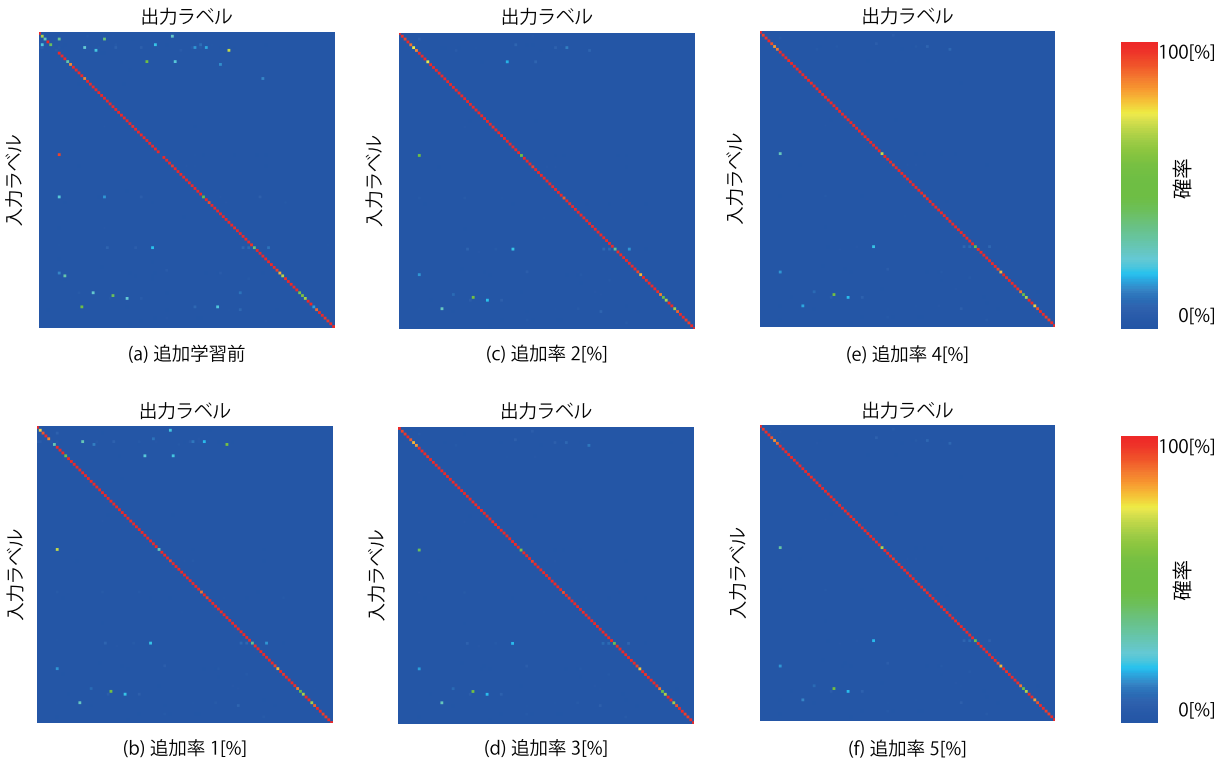

図 10　各手法の学習後の Confusion Matrix

4.2　人検出実験

次に，2 クラスの人検出問題でも提案手法が有効であるか評価実験を行う．識別精度の評価には Detection Error Trade-off(DET) カーブを用いる．DET カーブは横軸に False positive per window，縦軸に Miss rate を表しており，原点に近いほど高精度であることを表す．比較手法として，SVM と比較する．

4.2.1　データセット

データセットには INRIA Person Dataset[15] を用いる．学習用サンプルにはポジティブ 2,416 枚，ネガティブ 13,161 枚を使用する．検証用サンプルにはポジティブ 566 枚，ネガティブ 2,640,461 枚を使用し，評価用サンプルはポジティブ 556 枚，ネガティブサンプル 2,583,544 枚を使用する．

4.2.2　特徴量

特徴量には，人検出に多く利用されている Histograms of Oriented Gradients(HOG) 特徴量 [16] を用いる．HOG 特徴量は，画像をセルと呼ばれる局所領域毎に勾配方向ヒストグラムを作成し，複数のセルを正規化することで，照明変化や幾何学的変化に対する不変性を獲得している．本研究で用いる画像は 128 × 64 画素であり，セルを 8 ピクセル，ブロックを 2 セル，勾配方向を 9 方向とし算出される HOG 特徴量は 3780 次元となる．

4.2.3　実験結果

検証用サンプルに対して Random Forest による未検出サンプルは 30 枚存在し，誤検出サンプルは 41983 枚存在した．その中から重複せず，1%-3%をランダムサンプリングし，追加学習サンプルとした．図 12 に DET カーブを示す．図より，Random Forest に誤識別サンプルを追加し学習することで識別性能が向上している．

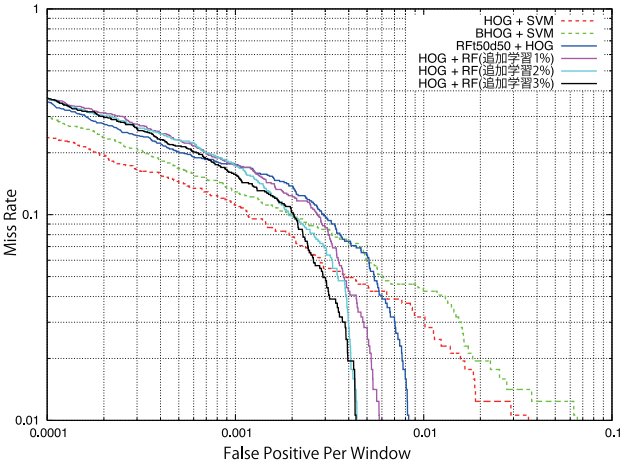

図 12　DET カーブ

5 おわりに

本稿では，Random Forest における追加学習を提案した．追加学習は，実利用シーンにおいて誤識別するサンプルを抑制する方法として有効である．その枠組みを Random Forest において実現した．Random Forest では，末端ノードを更新しても影響を受けるサンプルは少数であり，学習サンプルに対する影響は少ない．評価実験では，マルチクラス，2 クラス問題において評価し，その有効性を確認した．また，Vote Entropy による誤識別サンプルの自動判別では，目視による判別よりも追加学習のコスト削減を実現した．

参考文献

[1] B. Settles, “Active learning literature survey”, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.

[2] T. Scheffer, C. Decomain, and S. Wrobel, “Active Hidden Markov Models for Information Extraction”, In Proceedings of the International Conference on Advances in Intelligent Data Analysis (CAIDA), pp. 309-318, Springer Verlag, 2001.

[3] A. Holub, P. Perona, and M. Burl., “Entropy-based active learning for object recognition”, In CVPR, Workshop on Online Learning for Classification, 2008.

[4] I. Dagan, and Sean P. Engelson. “Committee-based sampling for training probabilistic classifiers”, ICML. Vol. 95. 1995.

[5] L. Breiman, “Random Forests”, Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[6] L. Breiman, “Bagging predictors”, Machine Learning, vol. 26, no. 2, pp. 123-140, 1996.

[7] Ho, T. K., “The random subspace method for constructing decision forests”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20 no. 8, pp. 832-844, 1998.

[8] J. Shotton, M. Johnson, and R. Cipolla, “Semantic texton forests for image categorization and segmentation”, in Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[9] Y. Amit and D. Geman, “Shape quantization and recognition with randomized trees”, Neural Comput., vol. 9, no. 7, pp. 1545-1588, 1997.

[10] V. Lepetit and P. Fua, “Keypoint recognition using randomized trees”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 9, pp. 1465-1479, 2006.

[11] Gall, J. and Yao, A. and Razavi, N. and Van Gool, L. and Lempitsky, V., “Hough forests for object detection, tracking, and action recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.33, no.11, pp.2188-2202, 2011.

[12] J. Shotton, and A. Fitzgibbon, and Cook, M. and Sharp, T. and Finocchio, M. and Moore, R. and Kipman, A. and Blake, A., “Real-time human pose recognition in parts from single depth images”, Computer Vision and Pattern Recognition, 2011.

[13] D. Tang, Y. Liu, and T.-K. Kim, “Fast pedestrian detection by cascaded random forest with dominant orientation templates”, in BMVC, pp. 1-11, 2012.

[14] P. Viola, and M. Jones. “Rapid object detection using a boosted cascade of simple features”, Computer Vision and Pattern Recognition, 2001.

[15] “INRIA Person Dataset”, http://pascal.inrialpes.fr/data/human/.

[16] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection”, in Computer Vision and Pattern Recognition, vol. 1, pp. 886-893, 2005.