

### 1. はじめに

2クラス識別の線形 SVM を用いた物体検出の高速化手法として、ベクトル分解法を用いた早期判定 [1] が提案されている。本手法は、重みベクトルを分解することで二値ベクトル間の演算となり高速化を実現しているが、多クラス識別に対応していない。そこで、本研究は行列分解法を提案し、早期棄却による多クラス識別の高速化を目的とする。

### 2. ベクトル分解法による識別の高速化

2 クラスの線形 SVM の識別関数は、重みベクトルを  $\mathbf{w} \in \mathbb{R}^D$ 、入力特徴ベクトルを  $\mathbf{x} \in \{-1, 1\}^D$  とすると、 $f(\mathbf{x}) = \text{sign}[\mathbf{w}^T \mathbf{x} + b]$  となる。ここで、二値ベクトル  $\mathbf{x}$  と実数ベクトル  $\mathbf{w}$  間の内積は計算コストが高い。Hare が提案したベクトル分解法 [2] を用いると、重みベクトルを二値基底行列  $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k\} \in \{-1, 1\}^{D \times k}$  とスケール係数ベクトル  $\mathbf{c} = \{c_1, c_2, \dots, c_k\} \in \mathbb{R}^k$  に分解することができる。

$$f(\mathbf{x}) = \text{sign}[\mathbf{w}^T \mathbf{x} + b] \tag{1}$$

$$\approx \text{sign}[\mathbf{c}^T \mathbf{M}^T \mathbf{x} + b] = \text{sign}\left[\left(\sum_{i=1}^k c_i \mathbf{m}_i^T \mathbf{x}\right) + b\right] \tag{2}$$

ここで  $k$  は分解の基底数を表す。これにより、二値ベクトル間の内積計算に置き換えることができ、高速な演算が期待できる。

本研究では、多クラスを対象とするため、one-vs.-rest による多クラス識別器を、クラス数  $J$  個の 2 クラス識別器で実現する。このとき、 $J$  個の重みベクトルからなる重み行列  $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_J\}^T$  をベクトル分解法で分解すると、図 1(a) のように  $J$  個の二値基底行列  $\mathbf{M}$  とスケール係数ベクトル  $\mathbf{c}$  が算出される。従って、多クラス識別では、特徴ベクトル  $\mathbf{x}$  と  $\mathbf{cM}$  の内積の計算回数が  $J$  回に増加するという問題がある。

### 3. 実数行列の分解による多クラス識別の高速化

重みベクトル  $\mathbf{w}$  の相関が高いとき、図 1(a) のように各二値基底行列  $\mathbf{M}$  においても類似した値を持つ。このとき、各二値基底行列  $\mathbf{M}$  を 1 つの行列で表現すると、 $\mathbf{x}$  と  $\mathbf{CM}$  の内積の近似計算が 1 回に減少する。そこで、本研究では図 1(b) のように、重み行列  $\mathbf{W}$  を二値基底行列  $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k\} \in \{-1, 1\}^{D \times k}$  と実数スケール行列  $\mathbf{C} = \{c_1, c_2, \dots, c_k\}^T \in \mathbb{R}^{J \times k}$  に分解する行列分解法を提案する。さらに、内積の近似計算処理においてカスケード構造による早期棄却を導入することで、識別の高速化を実現する。

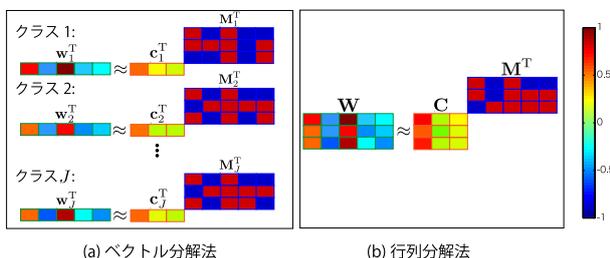


図 1：重み行列  $\mathbf{W}$  の分解

#### 3.1. 行列分解法

重み行列  $\mathbf{W}$  を二値基底行列  $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k\} \in \{-1, 1\}^{D \times k}$  とスケール係数行列  $\mathbf{C} = \{c_1, c_2, \dots, c_k\}^T \in \mathbb{R}^{J \times k}$  に分解する。式 (3) を最小化するように二値基底行列  $\mathbf{M}$  とスケール係数行列  $\mathbf{C}$  を最適化する。この分解法は、二値基底行列  $\mathbf{M}$  とスケール係数行列  $\mathbf{C}$  を交互に最適化する。提案する行列分解法の流れを Algorithm 1 に示す。

$$\|\mathbf{W} - \mathbf{CM}^T\|_F^2 \tag{3}$$

#### 3.2. 内積の近似計算の早期棄却による識別の高速化

行列分解において、重み行列  $\mathbf{W}$  と特徴ベクトル  $\mathbf{x}$  の内積計算による識別精度を維持するには、基底数  $k$  の値を大

### Algorithm 1 行列分解のアルゴリズム

```

Require:  $\mathbf{W}, L$ 
for  $l = 1$  to  $L$  do
     $\mathbf{M}_l$  を  $\{-1, 1\}$  の乱数により初期化.
     $\mathbf{C}_l$  を実数の乱数により初期化.
    repeat
         $\mathbf{M}_l$  を固定し、式 (3) を最小化する  $\mathbf{C}_l$  について最小二乗法により求める
         $\mathbf{C}_l$  を固定し、式 (3) を最小化する  $\mathbf{M}_l$  について  $\{-1, 1\}$  を総当たりし、最適化する.
    until 式 (3) の値が収束
end for
式 (3) が最小となる  $\mathbf{M}_l$  と  $\mathbf{C}_l$  を採用
return  $\{\mathbf{M}_l\}, \{\mathbf{C}_l\}$ 
    
```

きくする必要がある。しかし  $k$  の数が大きくなると  $\mathbf{cm}^T \mathbf{x}$  の演算回数が増えるため、高速な識別ができない場合がある。そこで、図 2 のように行列分解で算出した  $N$  個の二値基底行列  $\mathbf{M}$  とスケール係数行列  $\mathbf{C}$  をカスケード状に並び、非対象クラスの早期棄却を導入する。各段で、しきい値  $T$  を基準に非対象クラスを棄却する。

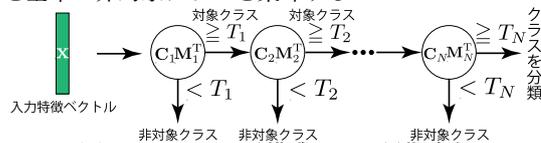


図 2：カスケード構造による早期棄却

### 4. 評価実験

提案手法の有効性を確認するために評価実験を行う。多クラスにおけるベクトル分解法と提案手法の行列分解法による誤差を比較する。実験に用いる識別器は SVM(one-vs.-rest) とし、学習には 4 クラスの標識画像から抽出した 288 次元の B-HOG 特徴量を使用する。また、17,024 枚の識別時間を比較する。クラスあたりの基底数に対する二乗誤差を図 3 に示す。

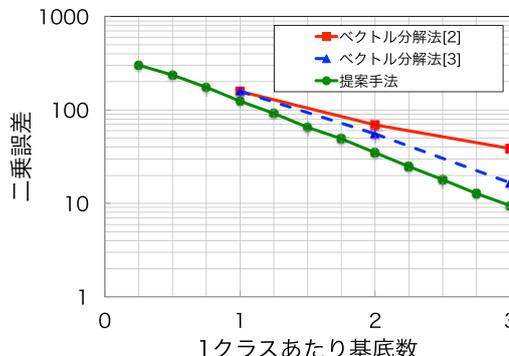


図 3：分解による誤差

図 3 より、1 クラスあたりの基底数が 3 のとき、提案手法はベクトル分解法 [3] より誤差を 43.1% 小さくすることができた。識別時間は、分解無しが 57.3[ms]、提案手法が 5.2[ms]、提案手法 + カスケードが 3.3[ms] となった。提案手法は分解前に比べ約 11 倍高速に、さらにカスケード構造を導入することで約 1.6 倍高速化することができた。

### 5. おわりに

提案手法は、行列分解法とカスケード構造による早期棄却を導入することで、高速な物体検出が可能であることを確認した。今後は行列分解法に適した識別の早期判定について検討する予定である。

### 参考文献

- [1] 後藤等, “近似計算を導入した線形識別器の早期判定による高速な識別” 信学論 D, pp. 294-302, vol. 97, No. 2, 2014.
- [2] S. Hare et al., “Efficient Online Structured Output Learning for Keypoint-Based Object Tracking”, CVPR, pp.1894-1901, 2012.
- [3] Y. Yamauchi et al., “Distance Computation Between Binary Code and Real Vector for Efficient Keypoint Matching”, IPSJ Trans. on CVA, pp.124-128, vol. 5, 2013.